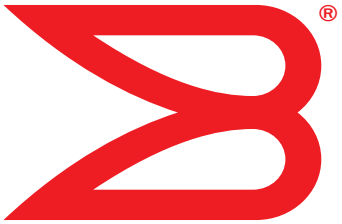
53-1002334-01
30 May 2012



ServerIron Traffic Works

Switching and Routing Guide

Supporting ServerIron TrafficWorks version 10.2.02

BROCADE

Copyright © 2012 Brocade Communications Systems, Inc. All Rights Reserved.

Brocade, Brocade Assurance, the B-wing symbol, DCX, Fabric OS, MLX, SAN Health, VCS, and VDX are registered trademarks, and AnyIO, Brocade One, CloudPlex, Effortless Networking, ICX, NET Health, OpenScript, and The Effortless Network are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. Other brands, products, or service names mentioned may be trademarks of their respective owners.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. shall have no liability or responsibility to any person or entity with respect to any loss, cost, liability, or damages arising from the information contained in this book or the computer programs that accompany it.

The product described by this document may contain “open source” software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit

http://www.brocade.com/support/oscd.

Brocade Communications Systems, Incorporated

Corporate and Latin American Headquarters
Brocade Communications Systems, Inc.
130 Holger Way
San Jose, CA 95134
Tel: 1-408-333-8000
Fax: 1-408-333-8101
E-mail: info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems China HK, Ltd.
No. 1 Guanghua Road
Chao Yang District
Units 2718 and 2818
Beijing 100020, China
Tel: +8610 6588 8888
Fax: +8610 6588 9999
E-mail: china-info@brocade.com

European Headquarters
Brocade Communications Switzerland Sàrl
Centre Swissair
Tour B - 4ème étage
29, Route de l'Aéroport
Case Postale 105
CH-1215 Genève 15
Switzerland
Tel: +41 22 799 5640
Fax: +41 22 799 5641
E-mail: emea-info@brocade.com

Asia-Pacific Headquarters
Brocade Communications Systems Co., Ltd. (Shenzhen WFOE)
Citic Plaza
No. 233 Tian He Road North
Unit 1308 – 13th Floor
Guangzhou, China
Tel: +8620 3891 2000
Fax: +8620 3891 2111
E-mail: china-info@brocade.com

Contents

CHAPTER 1
ABOUT THIS GUIDE 1-1

AUDIENCE 1-1
CONVENTIONS 1-1
RELATED DOCUMENTATION 1-1
GETTING TECHNICAL HELP 1-2
DOCUMENT FEEDBACK 1-2

CHAPTER 2
SWITCHING AND ROUTING 2-1

MAC SWITCHING 2-1
- STATIC MAC ENTRIES 2-2
- DISPLAYING MAC ADDRESSES 2-2
- DISPLAYING MAC ADDRESS STATISTICS 2-3

STP 2-3
- CONFIGURING FAST PORT SPAN 2-3
- CONFIGURING FAST UPLINK SPAN 2-3
- MODIFYING SPANNING TREE PARAMETERS 2-4
- DISPLAYING SPANNING TREE STATISTICS 2-5
- IRONSPAN STP ENHANCEMENTS 2-6

TRUNK GROUPS 2-6
- 802.3AD LINK AGGREGATION 2-6

VLANS 2-6
- PORT-BASED VLANS 2-6
- CHANGING THE TAG TYPE 2-7
- ENABLING AGGREGATED VLAN 2-7
- CHANGING THE ID OF THE DEFAULT VLAN 2-7
- ALLOWING TFTP ACCESS ONLY TO CLIENTS IN A SPECIFIC VLAN 2-7
- DISABLING OR RE-ENABLING DYNAMIC DISCOVERY OF PROTOCOL VLANS 2-8
- SETTING THE MAXIMUM NUMBER OF VLANS 2-8

 © 2012 Brocade Communications Systems, Inc.

Assigning Tagged or Untagged Ports to a Port-Based VLAN 2-8
Configuring Uplink Ports 2-8
Setting a Priority for a VLAN 2-9
Configuring an AppleTalk Protocol VLAN 2-9
Configuring a DECnet Protocol VLAN 2-9
Configuring an IP Protocol VLAN 2-10
Configuring an IP Subnet Protocol VLAN 2-10
Configuring an IPX Network VLAN 2-10
Configuring an IPX Protocol VLAN 2-11
Configuring a NetBIOS Protocol VLAN 2-11
Configuring Another Protocol VLAN 2-12
Dual-Mode VLAN Ports 2-12
MAC Filters 2-14
Configuring Filters for Layer 2 Filtering 2-14
Additional Examples of Layer 2 MAC Filter Definitions 2-16
Abbreviating the Address or Mask.... 2-16
Setting the MAC Age Time 2-16
Enabling Logging of Packets Denied by Layer 2 MAC Filters 2-17
Address-Lock Filters 2-17
Configuring a Broadcast Filter 2-17
Setting the Broadcast Limit 2-18
Assigning a Gateway List 2-18
Multicast 2-19
Enabling IP Multicast Traffic Reduction 2-19
Configuring a Multicast Filter 2-19
Setting the Multicast Limit 2-20
Disabling IGMP Queries 2-20
Adding an IP Interface 2-20
Adding a Static IP Route 2-22
Adding a Static ARP Entry 2-22
Clearing the ARP Cache 2-23
Clearing the IP Cache 2-23
Clearing the MAC Address Table 2-23
Setting System Max 2-23
Adding a Static MAC Address 2-24
Displaying IP Forwarding Information 2-25
Displaying IP Forwarding State Information 2-25
Displaying the IP Host Table 2-26
Displaying the ARP Cache or the Static ARP Table 2-26
Displaying Static ARP Entries 2-28
Displaying a List of IP Interfaces 2-28
Displaying the IP Route Table 2-29
Displaying IP Forwarding Traffic Statistics 2-30
Clearing IP Traffic Statistics 2-31
IP Interfaces and Multinetting 2-31
The source-nat Parameter.... 2-32
Disabling Layer 2 Switching 2-32

 © 2012 Brocade Communications Systems, Inc.

CONFIGURING A DECNET PROTOCOL VLAN 2-32
CONFIGURING AN IP INTERFACE 2-32
CONFIGURING AN IP FILTER 2-33
SLB EXAMPLE 2-34
TCS USES OF FILTERS 2-34
POLICY-BASED CACHE SWITCHING 2-35
SETTING THE TTL 2-35
CONFIGURING AN IP PROTOCOL VLAN 2-35
CONFIGURING AN IP SUBNET PROTOCOL VLAN 2-35
RIP 2-36
ENABLING RIP 2-36
RIP TIMERS 2-37
REDISTRIBUTING IP STATIC ROUTES INTO RIP 2-37
ENABLING REDISTRIBUTION 2-38
DENYING REDISTRIBUTION 2-38
PERMITTING REDISTRIBUTION 2-39
LEARNING RIP DEFAULT ROUTES 2-40
ENABLING POISON REVERSE OR SPLIT HORIZON 2-40
OSPF 2-40
DYNAMIC LINK AGGREGATION 2-41
CONFIGURATION RULES 2-41
VALID AGGREGATE LINKS 2-42
FLEXIBLE TRUNK ELIGIBILITY 2-44
ENABLING LINK AGGREGATION 2-45
ENABLING LINK AGGREGATION AND USING THE DEFAULT KEY 2-45
ASSIGNING A UNIQUE KEY AND ENABLING LINK AGGREGATION 2-46
CONFIGURING KEYS FOR PORTS WITH LINK AGGREGATION ENABLED 2-46
LINK AGGREGATION PARAMETERS 2-47
SYSTEM PRIORITY 2-47
PORT PRIORITY 2-47
LINK TYPE 2-47
KEY 2-47
ABOUT BLOCKED PORTS 2-49
DISPLAYING AND DETERMINING THE STATUS OF AGGREGATE LINKS 2-50
CLEARING THE NEGOTIATED LINK AGGREGATIONS 2-52

 © 2012 Brocade Communications Systems, Inc.

 © 2012 Brocade Communications Systems, Inc.

Chapter 1
About this Guide

This guide describes the switching and routing features of the Brocade® ServerIron devices.

Audience

This guide is intended for network engineers with a basic knowledge of switching, routing, and application traffic management.

Conventions

This guide uses the following typographical conventions to describe information:

Italic	Highlights the title of another publication or emphasizes a word or phrase.
`Bold code`	Indicates code that is entered exactly as shown.
Bold	Indicates a command or keyword that can be entered exactly as is.

NOTE: A note emphasizes an important fact or calls your attention to a dependency.

WARNING: A warning calls your attention to a possible hazard that can cause injury or death.

CAUTION: A caution calls your attention to a possible hazard that can damage equipment.

Related Documentation

For more information, refer to the following Brocade Communications Systems ServerIron documentation:

- *Release Notes for ServerIron Switch and Router Software TrafficWorks 10.2.00* – provides a list of new features and enhancements, upgrade procedures, and bug fixes.
- *ServerIron TrafficWorks Graphical User Interface* – provides details on the graphical user interface for the

© 2012 Brocade Communications Systems, Inc.

ServerIron family of application delivery controllers.

- *ServerIron TrafficWorks Server Load Balancing Guide* – describes basic Server Load Balancing configurations for the ServerIron product family. It covers the following features: Server Load Balancing, Stateless Server Load Balancing, Health Checks, Layer 7 Content Switching, and High Availability
- *ServerIron TrafficWorks Advanced Server Load Balancing Guide* – discusses Advanced Server Load Balancing concepts for the ServerIron product family. It covers the following features: are SIP Server Load Balancing, Transparent Cache Switching, IDS Server Load Balancing, HTTP Compression, and Total Content Analysis
- *ServerIron TrafficWorks Global Server Load Balancing Guide* – explains how one can achieve site level redundancy and data center site failure protection using Global Server Load Balancing feature of ServerIron
- *ServerIron TrafficWorks Security Guide* – describes Security features of ServerIron product family. It covers the following features: are Secure Socket Layer (SSL) Acceleration, Web Application Firewall, Deep Packet Scan, Access Control List, and Network Address Translation
- *ServerIron TrafficWorks Administration Guide* – discusses different administrative configurations for the ServerIron product family.
- *ServerIron TrafficWorks Switching and Routing Guide* – describes switching and routing configurations on the ServerIron product family
- Brocade *ServerIron Firewall Load Balancing Guide* – provides detailed feature descriptions, procedures, and application examples for Firewall Load Balancing.
- Brocade *ServerIron Chassis Hardware Installation Guide* – provides the physical characteristics, power consumption, and performance capabilities of the ServerIron chassis switch families, and explains how to set up and install the switches and their modules.
- Brocade *Management Information Base Reference* – presents the Simple Network Management Protocol (SNMP) Management Information Base (MIB) objects that are supported on Brocade devices.

The latest version of these guides are posted at http://www.brocade.com/ethernetproducts.

If you find errors in the guides, send an e-mail to documentation@brocade.com

Getting technical help

To contact Technical Support, go to *http://www.brocade.com/services-support/index.page* for the latest e-mail and telephone contact information.

Document feedback

Quality is our first concern at Brocade and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. Forward your feedback to:

documentation@brocade.com

Provide the title and version number of the document and as much detail as possible about your comment, including the topic heading and page number and your suggestions for improvement.

 © 2012 Brocade Communications Systems, Inc.

Chapter 2
Switching and Routing

This chapter describes Layer 2 switching and routing for ServerIron devices. It contains the following sections:

- “MAC switching” on page 2-1
- “STP” on page 2-3
- “Trunk Groups” on page 2-6
- “VLANs” on page 2-6
- “MAC Filters” on page 2-14
- “Address-Lock Filters” on page 2-17
- “Configuring a Broadcast Filter” on page 2-17
- “Setting the Broadcast Limit” on page 2-18
- “Assigning a Gateway List” on page 2-18
- “Multicast” on page 2-19

MAC switching

All Brocade devices support MAC switching. MAC switching enables intelligent wire-speed bridging of Layer 2 packets. The first time a Brocade device receives a packet from a given MAC destination, the device makes an entry in its Layer 2 cache. The entry consist of the packet’s source MAC address and the port on which the device received the packet.

When the device receives a bridge packet destined for the cached address, the device does not need to send the packet as a broadcast through all the ports within the broadcast domain. Instead, the device can intelligently send the packet only through the port to which the destination device is connected. Thus, even though Layer 2 domains are typically broadcast domains, MAC switching enhances performance in the domain by reducing the amount of broadcast traffic in the domain.

In addition, Brocade routers that are enabled for MAC switching can switch traffic for route protocols that are not supported in the routing software. If IPX routing is disabled on a router, the router can switch the IPX packets instead.

To avoid accumulating stale cache entries, Brocade devices use an aging mechanism. The aging mechanism removes a learned entry from the cache after the entry has remained unused for a specified interval (by default, 300 seconds). You can change or disable the aging interval.

© 2012 Brocade Communications Systems, Inc.

By default, all ports in a Brocade device belong to a common Layer 2 broadcast domain, VLAN 1. You can configure port-based VLANs (Virtual LANs) to create smaller broadcast domains that use subsets of the device's ports.

Static MAC Entries

MAC entries that the Brocade device learns and caches are subject to an aging time. After a cached entry remains unused for the duration of the aging time, the software removes the entry from the Layer 2 cache. If you want certain MAC addresses to always be present in the device's Layer 2 address table, you can add them as static entries.

A static MAC entry, like a cached (dynamic) MAC entry, maps a MAC address to the Brocade device's port attached to that device.

Unlike cached MAC entries, static MAC entries provide the following benefits:

- You can assign a QoS priority to a static MAC entry.
- You can specify VLAN membership for a static MAC entry.
- A static entry prevents broadcast storms that can be caused when a server's MAC entry is removed. For example, if a server goes down long enough for the server's entry to age out, the Brocade device sends packets addressed to the server as broadcasts until the device relearns the cache entry for the server.

You can specify port priority (QoS) and VLAN membership (VLAN ID) for the MAC address. On switches, you also can specify the device type (router or host) for the entry.

NOTE: On Brocade routers, you also can create static IP routes, ARP entries, and RARP entries. The ServerIron and other Brocade switches support only static MAC addresses.

Displaying MAC Addresses

To display all MAC addresses on a ServerIron, enter the following:

```
ServerIron(config)# show mac-address
Total entries from all ports = 75
       MAC      Port   Age CamF CIDX0 CIDX1 CIDX2 CIDX3 CIDX4 CIDX5
0000.0300.0000  10  17293 00H     0     0     0     0     0     0
0060.089f.8086   1     12 0bH    23    15     0     6     0     0
0060.9709.914b  16   2130 00H     0     0     0     0     0     0
00a0.249a.0163  16    130 00H     0     0     0     0     0     0
0060.979d.41a5  11    475 00H     0     0     0     0     0     0
00a0.24c5.01d1  11      0 0cH     0     0    20    14     0     0
0060.979d.41df  11    570 00H     0     0     0     0     0     0
0060.9759.4226  16    240 00H     0     0     0     0     0     0
0060.9759.4235  16    130 00H     0     0     0     0     0     0
0800.208f.725b   2    135 00H     0     0     0     0     0     0
0060.9759.4264  16      0 0aH     0    14     0    21     0     0
00a0.24c5.02a1  16     15 09H     5     0     0    33     0     0
0000.c02c.a2bf   7     11 03H    27     5     0     0     0     0
00a0.24c5.02f8   4    135 00H     0     0     0     0     0     0
00a0.24c5.02fc   6      0 06H     0     8    31     0     0     0
0800.207e.c312   2      2 0dH    25     0    24    13     0     0
0800.208f.5331   2    135 00H     0     0     0     0     0     0
00e0.5200.0385  10   5160 00H     0     0     0     0     0     0
--More--, next page: Space/Return key, quit: Control-c
```


© 2012 Brocade Communications Systems, Inc.

NOTE: The information displayed in columns with headings **CamF**, and **CIDX0** through **CIDX5**, is not relevant for day-to-day management of the ServerIron. The information is used by engineering and technical support staff for debug purposes.

Syntax: show mac-address [ethernet <portnum> | <mac-addr> | session]

The **session** keyword causes information about MAC session entries to be displayed.

Displaying MAC Address Statistics

To display the total number of MAC addresses currently active on a ServerIron, enter the following command:

```
ServerIron(config)# show mac-address-statistics
Total entries  = 41
Port     1     2     3     4     5     6     7     8     9
         0     6    11     1     1     1     2     1     1
Port     10     11      12    13    14    15    16
          0      3       1     3     1     1     8
```

For each port, the number of learned MAC addresses is displayed.

Syntax: show mac-address-statistics

This command serves as a numerical summary of the detailed summary provided by the command **show mac-addresses**.

STP

The Spanning Tree Protocol (STP) detects and eliminates logical loops in a Layer 2 broadcast domain. STP is described in the IEEE 802.1d bridge protocols standard and ensures the device uses the most efficient path when multiple paths exist between ports. If a selected path fails, STP searches for and then establishes an alternate path to prevent or limit retransmission of data.

For ServerIron devices, STP is disabled by default on Router (R) code images but enabled by default on the Switch (S) code images.

Configuring Fast Port Span

The Fast Port Span feature allows faster STP convergence on ports that are attached to end stations. To configure this feature, enter the following command:

```
ServerIron(config)#fast port-span
```

To exclude a port from Fast Port Span, while leaving Fast Port Span enabled globally, enter the following command:

```
ServerIron(config)#fast port-span exclude ethernet 1
```

Syntax: [no] fast port-span [exclude ethernet <portnum> [ethernet <portnum>… | to <portnum>]

Configuring Fast Uplink Span

The Fast Uplink Span feature reduces the convergence time for uplink ports to another device to just four seconds (two seconds for listening and two seconds for learning). To configure this feature, enter the following command:

```
ServerIron(config)# fast uplink-span ethernet 1 to 4
```

Syntax: [no] fast uplink-span [ethernet <portnum> [ethernet <portnum>… | to <portnum>]

 © 2012 Brocade Communications Systems, Inc.

Replace the <portnum> with a port that has redundant uplinks on a wiring closet switch.

Modifying Spanning Tree Parameters

Spanning Tree bridge and port parameters are configurable using the **spanning-tree** command. When no port-based VLANs are active on the system, spanning tree parameters are set at the Global CONFIG Level.

When port-based VLANs are active on the system, spanning tree protocol bridge and port parameters can be configured globally at the VLAN Level. Additionally, you can disable or enable STP on an interface basis.

NOTE: If VLANs are active on a switch or router, spanning-tree will not be seen as an option at the Global CONFIG Level of the CLI but will be an option of the VLAN Level.

All bridge and port parameters have default values and do not need to be modified unless required to match network needs. Additionally, all values will be globally applied to the switch or router. By default this feature is enabled on switches and disabled on routers.

You can modify the following STP Parameters:

1. Modify bridge parameters—forward delay, maximum age, hello time and priority
2. Modify port parameters—priority and path cost

Suppose you want to enable spanning tree on a system in which no port-based VLANs are active and change the hello-time from the default value of 2 to 8 seconds. Additionally, suppose you want to change the path and priority costs for port 5 only. To do so, enter the following commands.

```
ServerIron(config)#span hello-time 8
ServerIron(config)#span ethernet 5 path-cost 15 priority 64
```

To disable spanning tree on physical port 4 of a system with no VLANs operating:

```
ServerIron(config)#interface ethernet 4
ServerIron(config-if-4)no spanning-tree
```

Syntax: [no] spanning-tree [ethernet <portnum> path-cost <value> priority <value>] forward-delay <value> hello-time <value> maximum-age <time> priority <value>

Bridge Parameters:

- forward-delay: Possible values: 4 – 30 seconds. Default is 15 seconds.
- max-age: Possible values: 6 – 40 seconds. Default is 20 seconds.
- hello-time: Possible values: 1 – 10 seconds. Default is 2 seconds.
- priority: Possible values: 1 – 65,535. Default is 32,768. A higher numerical value means a lower priority; thus, the highest priority is 0.

Port Parameters:

- path: Possible values: 1-65,535. Default: Auto
- The default value ‘Auto’ means that the port will adjust the default value automatically based on the port speed. The default value is based on the following formula: Half-duplex ports: 1000/port speed; Full-duplex ports: (1000/port speed)/2
- Priority: possible values are 0-255. Default is 128. A higher numerical value means a lower priority; thus, the highest priority is 0.

 © 2012 Brocade Communications Systems, Inc.

Displaying Spanning Tree Statistics

To display spanning tree statistics, enter the following command:

```
ServerIron#show span ?
  DECIMAL     Number of spanning tree entries to skip before display begins
  detail      Show more details of STP information on each port
  pvst-mode   PVST status
  vlan        Show spanning tree of a VLAN
  |           Output modifiers
  <cr>
ServerIron#show span

VLAN 1 BPDU cam_index is 2061 and the DMA master Are(HEX) 4
STP instance owned by VLAN 1
Global STP (IEEE 802.1D) Parameters:

VLAN Root             Root Root Prio Max He- Ho- Fwd Last     Chg  Bridge
 ID   ID              Cost Port rity Age llo ld  dly Chang    cnt  Address
                           Hex  sec sec sec sec sec
   1 8000000cdb2bad20 0    Root 8000 20  2   1   15  1456133  0    000cdb2bad20

Port STP Parameters:

Port  Prio Path  State      Fwd    Design   Designated       Designated
Num   rity Cost             Trans  Cost     Root             Bridge
      Hex
2/1   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/2   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/3   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/4   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/5   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/6   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/7   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/8   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/9   80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/10  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/11  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/12  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/13  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/14  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/15  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/16  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/17  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/18  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/19  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/20  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/21  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/22  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/23  80   0     DISABLED   0      0        0000000000000000 0000000000000000
2/24  80   0     DISABLED   0      0        0000000000000000 0000000000000000
```

 © 2012 Brocade Communications Systems, Inc.

IronSpan STP Enhancements

IronSpan is a set of Layer 2 features that extend the operation of standard STP. IronSpan enables you to fine tune standard STP and avoid some of its limitations. IronSpan includes the following features:

- Fast Port Span – By default, devices running Fast Port Span perform Spanning Tree Protocol (STP) convergence in four seconds instead of 30 or more seconds for certain ports connected to end stations.
- Fast Uplink Span – Enhances STP by allowing a Brocade device with redundant uplinks to quickly resume forwarding, in just four seconds. This feature is similar to Fast Port Span but applies to certain inter-switch links on Brocade devices, instead of Brocade links to end stations.

Trunk Groups

A trunk group is a set of ports that provide a high speed link between two Brocade devices or between a Brocade device and a server. A trunk group can consist of up to four ServerIron physical ports and provides the bandwidth of those ports combined. Thus, a trunk group containing four 1 Gbps ports can provide up to four Gbps of bi-directional traffic. See the **trunk server** global command.

In addition to enabling load sharing of traffic, trunk groups provide redundant, alternate paths for traffic. Thus, if a link in a trunk group fails, the device still uses the other links in the trunk group.

With Release 7.1.01 and later, you can configure up to 12 trunks per 24 ports.

802.3ad Link Aggregation

802.3ad is a standards-based approach for aggregating several switch ports. With Release 09.5.02a, the ServerIron now supports standards-based 802.3ad LACP link aggregation. This feature allows you to connect the ServerIron to devices from other vendors through port-aggregated channels.

VLANs

By default, all ports and interfaces running a Switch (S) image are kept in VLAN 1, which is named DEFAULT VLAN:

```
!
vlan 1 name DEFAULT-VLAN by port
!
```

Use **show vlan** to display interesting information:

```
ServerIron#show vlan
Total PORT-VLAN entries: 1
Maximum PORT-VLAN entries: 32

legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree On
 Untagged Ports: (S2)  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16
 Untagged Ports: (S2) 17 18 19 20 21 22 23 24
   Tagged Ports: None
   Uplink Ports: None
```

Port-Based VLANs

By default, all ports in a Brocade device belong to a common Layer 2 broadcast domain. When the device sends a broadcast packet, the packet goes out all active ports. A port-based Virtual LAN (VLAN) is a subset of ports on a Brocade device that constitutes a Layer 2 broadcast domain.

 © 2012 Brocade Communications Systems, Inc.

Port-based VLANs can reduce the likelihood and severity of broadcast storms by reducing the number of ports affected by a storm. In addition, for devices such as servers that can cause broadcast storms, you can add static MAC entries for the devices and assign the static entries to a VLAN.

Each port-based VLAN maintains a separate spanning tree. (See “STP” on page 2-3.)

Changing the Tag Type

Tag type is the value that will be sent out on a packet to indicate it as tagged VLAN port. The 802.1q standard recognizes the value of 8100 for this purpose. Other values can be assigned to this parameter but are not recommended.

Brocade switches support 802.1q VLAN tagging. VLAN tagging is a method of identifying a packet as a member of a VLAN. VLAN tagging enables you to configure ports on multiple switches into a single VLAN. Using tagged VLANs can ease network management and ensures interoperability with other devices.

When a switch sends a packet that is a member of a tagged VLAN, the switch "tags" the packet to indicate its VLAN membership. Other switches that support VLAN tagging recognize the tag and process the packet according to its VLAN membership.

```
ServerIron(config)#tag-type 8100
```

Syntax: [no] tag-type <value>

The default `<value>` is 8100.

Enabling Aggregated VLAN

A larger Ethernet frame size for VLAN aggregation changes the maximum Ethernet size to 1530 bytes.

To enable a larger Ethernet frame size for VLAN aggregation, enter the following command:

```
ServerIron(config)# aggregated-vlan
```

Syntax: [no] aggregated-vlan

Use this command when you are configuring Super Aggregated VLANs.

Changing the ID of the Default VLAN

Before you change the VLAN ID for the default VLAN, ensure that the ID is not already in use. For example, if you have already defined VLAN 10, do not try to use "10" as the new VLAN ID for the default VLAN. Valid VLAN IDs are numbers .

When you enable port-based VLAN operation, all ports are assigned to VLAN 1 by default. As you create additional VLANs and assign ports to them, the ports are removed from the default VLAN. All ports that you do not assign to other VLANs remain members of default VLAN 1. This behavior ensures that all ports are always members of at least one VLAN.

NOTE: Changing the default VLAN name does not change the properties of the default VLAN. Changing the name allows you to use the VLAN ID "1" as a configurable VLAN.

```
ServerIron(config)#default-vlan-id 1001
```

Syntax: [no] default-vlan-id <value>

The `<value>` is from 1 – 4095. The default is 1.

Allowing TFTP Access Only to Clients in a Specific VLAN

You can allow TFTP access only to clients in a specific VLAN. The following example configures the device to allow TFTP access only to clients connected to ports within port-based VLAN 40. Clients connected to ports that are not in VLAN 40 are denied access.

```
ServerIron(config)#tftp client enable vlan 40
```

 © 2012 Brocade Communications Systems, Inc.

Syntax: [no] tftp client enable vlan <vlan-id>

Disabling or Re-enabling Dynamic Discovery of Protocol VLANs

Dynamic discovery of protocol VLANs on switch-to-switch links enables switch-to-switch links to be automatically included in protocol VLANs that have dynamic port membership.

By default, the command is enabled. To disable dynamic discovery of protocol VLANs on switch-to-switch links, enter the following command:

```
ServerIron(config)#no vlan-dynamic-discovery
```

Syntax: [no] vlan-dynamic-discovery

Setting the Maximum Number of VLANs

You can adjust the maximum number of VLANs that are supported on a ServerIron using the following command:

```
ServerIron(config)# vlan max-vlan 128
```

Syntax: [no] vlan max-vlan <value>

The <value> variable can be set to a number in the following range: 1 – 4095. The default is 32.

Defining a large number of VLANs on a ServerIron reduces the number of VIPs possible because the system creates internal MAC addresses for VIPs. Creating a large number of VLANs can cause you to reach the maximum available MAC limit.

NOTE: Changing the maximum VLAN values using this command requires a system reload.

Assigning Tagged or Untagged Ports to a Port-Based VLAN

Once a port-based VLAN is created, port membership for that VLAN must be defined. To assign a port to a port-based VLAN, either the **tagged** or **untagged** command is used. When a port is tagged, it can be a member of multiple port-based VLANs.

When a port is tagged, it allows communication among the different VLANs to which it is assigned. A common use for this might be to place an email server that multiple groups may need access to on a tagged port, that in turn, is resident in all VLANs that members need access to the server.

Suppose you want to make port 5 (module 5), a member of port-based VLAN 4, a tagged port. Enter commands such as the following:

```
ServerIron(config)#vlan 4
ServerIron(config-vlan-4)#tagged ethernet 3/5
```

To assign all ports on a 16-port ServerIron except port 5 (module 3) as untagged to a VLAN, assign ports 1-4 and 6-16 to VLAN 4, enter commands such as the following:

```
ServerIron(config)#vlan 4
ServerIron(config-vlan-4)#untagged ethernet 3/1 to 3/4 e 3/6 to 3/16
```

Syntax: [no] tagged ethernet <portnum> [to <portnum> [ethernet <portnum>]]

Syntax: [no] untagged ethernet <portnum> [to <portnum> ethernet <portnum>]

Configuring Uplink Ports

When you configure a set of ports within a port-based VLAN as uplink ports for the VLAN, all broadcast and unknown-unicast traffic goes only to the uplink ports, not to the other ports in the VLAN.

To configure a port-based VLAN containing uplink ports, enter commands such as the following:

```
ServerIron(config)# vlan 10 by port
```

 © 2012 Brocade Communications Systems, Inc.

```
ServerIron(config-vlan-10)# untag ethernet 1/1 to 1/24
ServerIron(config-vlan-10)# untag ethernet 2/1 to 2/2
ServerIron(config-vlan-10)# uplink-switch ethernet 2/1 to 2/2
```

In this example, 24 ports on a 10/100 module and two Gigabit ports on a Gigabit module are added to port-based VLAN 10. The two Gigabit ports are then configured as uplink ports.

Syntax: [no] uplink-switch ethernet <portnum> [to <portnum> | ethernet <portnum>]

Setting a Priority for a VLAN

When you assign a higher priority to a VLAN so that in times of congestion, it will receive precedence over other transmissions. To do this, enter commands such as the following:

```
ServerIron(config)#vlan 25
ServerIron(config-vlan-25)#priority high
```

Syntax: [no] priority normal | high

Configuring an AppleTalk Protocol VLAN

You can create an AppleTalk protocol VLAN within a ServerIron port-based VLAN when entered at the VLAN Level. All ports are assumed by default to be members of the VLAN when initially created.

To create an AppleTalk Protocol VLAN with permanent port membership of 9 and 13 and no dynamic ports within an already defined port-based VLAN 2, enter commands such as the following:

```
ServerIron(config)#vlan 2
ServerIron(config-vlan-2)#atalk-proto
ServerIron(config-vlan-atalk-proto)#static e 9 e 13
ServerIron(config-vlan-atalk-proto)#no dynamic
```

If configuring this on a switch, enter **vlan 2 by port** at the CONFIG Level versus **vlan 2**, as shown in the example.

Protocol VLAN membership can be modified using the **dynamic**, **static**, or **exclude** commands.

To specify a VLAN name, use the name keyword followed by a string. The **name** keyword and string are the last arguments in the command. For example, to name an AppleTalk VLAN:

```
ServerIron(config)#atalk-proto name AppleVLAN1
```

To name an IP VLAN:

```
ServerIron(config)#ip-proto 192.75.5.0/24 name "Ship and Recv"
```

This example shows how to specify a name that contains a blank. Use double quotation marks before and after the name.

Syntax: [no] atalk-proto [<name>]

The <name> can be up to 16 characters long and can contain blanks.

Configuring a Decnet Protocol VLAN

You can create a Decnet protocol VLAN within a ServerIron port-based VLAN, when entered at the VLAN Level. All ports are assumed by default to be members of the VLAN when initially created.

To create a Decnet protocol VLAN with permanent port membership of 15 and 16 with port 17 as dynamic member port, within VLAN 5, enter the following commands.

```
ServerIron(config)#vlan 5
ServerIron(config-vlan-5)#decnet-proto
ServerIron(config-vlan-decnet-proto)#exclude e 1 to 14 e18
```

If configuring this on a switch, enter **vlan 5 by port** at the CONFIG Level versus **vlan 5**, as shown in the example above.

Protocol VLAN membership can be modified using the **dynamic**, **static**, or **exclude** commands.

 © 2012 Brocade Communications Systems, Inc.

Syntax: [no] decnet-proto [<name>]

To specify a VLAN name, use the name keyword followed by a string. The **name** keyword and string are the last arguments in the command. The name can contain blank spaces if you use double quotation marks before and after the name. The <name> can be up to 16 characters long and can contain blanks.

Configuring an IP Protocol VLAN

You can create an IP protocol VLAN on a ServerIron within a port-based VLAN, when entered at the VLAN Level. When configuring on a switch, all ports are dynamically allocated to the VLAN. You can modify port membership by using the **static** or **exclude** commands.

If configuring on a Brocade router, ports must be added to the VLAN with the **static** command. Ports are not dynamically allocated to IP protocol VLANs.

To assign ports 1, 2, 6 and 8 to an IP protocol VLAN within VLAN 7, enter commands such as the following:

```
ServerIron(config)#vlan 7
ServerIron(config-vlan-7)#ip-proto
ServerIron(config-vlan-ip-proto)#static e 1 to 2 e 6 e 8
```

If configuring this on a switch, enter **vlan 7 by port** at the CONFIG Level versus **vlan 7**, as shown in the example above.

Syntax: [no] ip-proto [<name>]

The <name> can be up to 16 characters long and can contain blanks.

An IP protocol and IP sub-net VLAN cannot both be configured to operate on a ServerIron at the same time. This restriction is also true for IPX and IPX network VLANs.

Configuring an IP Subnet Protocol VLAN

You can create an IP subnet protocol VLAN on a ServerIron within a port-based VLAN, when entered at the VLAN Level. This allows you to define additional granularity than that of an IP protocol VLAN, by partitioning the broadcast domains by sub-net. In creating an IP subnet VLAN, an IP address is used as identifier.

When configuring on a switch, all ports are dynamically allocated to the VLAN. You can modify port membership by using the **static** or **exclude** commands.

When configuring on a Brocade router, ports must be added to the VLAN with the **static** command. Ports are not dynamically allocated to IP subnet VLANs.

To create an IP sub-net of IP address 192.75.3.0 with permanent port membership of 1 and 2 (module 2), within VLAN 10, enter commands such as the following:

```
ServerIron(config)#vlan 10
ServerIron(config-vlan-10)#ip-subnet 192.75.3.0 255.255.255.0
ServerIron(config-vlan-ip-subnet)#static e 1 to 2
```

If configuring this on a switch, enter **vlan 10 by port** at the CONFIG Level versus **vlan 10**, as shown in the example.

Syntax: [no] ip-subnet <ip-addr> <ip-mask> [<name>]

The <name> can be up to 16 characters long and can contain blanks.

An IP protocol and IP sub-net VLAN cannot both be configured to operate simultaneously on a Brocade switch or router. This restriction is also true for IPX and IPX Network VLANs.

Configuring an IPX Network VLAN

You can create an IPX network VLAN on a ServerIron within a port-based VLAN, when entered at the VLAN Level.

This command you to define additional granularity than that of the IPX protocol VLAN, by partitioning the broadcast domains by IPX network number. In creating an IPX network VLAN, an IPX network number is used as identifier. The frame type must also be specified.

 © 2012 Brocade Communications Systems, Inc.

When configuring on a switch, all ports are dynamically allocated to the VLAN. You can modify port membership by using the **static** or **exclude** commands.

When configuring on a Brocade router, ports must be added to the VLAN with the **`static`** command. Ports are not dynamically allocated to IPX network VLANs.

To create an IPX network VLAN with a network number of 500 and frame type of 802.2 with permanent port membership of 10 and 14 within port-based VLAN 15, enter commands such as the following:

```
ServerIron(config)#vlan 15
ServerIron(config-vlan-15)#ipx-network 500 ethernet_802.2
ServerIron(config-vlan-ipx-proto)#static e 10 e 14
```

If configuring this on a switch, enter **vlan 15 by port** at the CONFIG Level versus **vlan 15**, as shown in the example above.

Syntax: [no] ipx-network <ipx-network-number> <frame-type> [<name>]

Possible <frame-type> values include ethernet_ii, ethernet_802.2, ethernet_802.3, and ethernet_snap. The <name> parameter can be up to 16 characters long and can contain blanks.

An IPX network and IPX protocol VLAN cannot both be configured to operate simultaneously on a Brocade switch or router. This restriction is also true for IP protocol and IP sub-net VLANs.

Configuring an IPX Protocol VLAN

You can create an IPX protocol VLAN on a ServerIron within a port-based VLAN, when entered at the VLAN Level.

When configuring on a switch, all ports are dynamically allocated to the VLAN. You can modify port membership by using the **static** or **exclude** commands.

If configuring on a Brocade router, ports must be added to the VLAN with the **static** command. Ports are not dynamically allocated to IPX protocol VLANs.

To assign ports 1, 2, 6 and 8 to an IPX protocol VLAN within port-based VLAN 22, enter commands such as the following:

```
ServerIron(config)#vlan 22
ServerIron(config-vlan-22)#ipx-proto
ServerIron(config-vlan-ipx-proto)#static e 1 to 2 e 6 e 8
```

If configuring this on a switch, enter **vlan 22 by port** at the CONFIG Level versus **vlan 22**, as shown in the example above.

Syntax: [no] ipx-proto [<name>]

The <name> can be up to 16 characters long and can contain blanks. To specify a VLAN name, use the name keyword followed by a string. The **name** keyword and string are the last arguments in the command. The name can contain blank spaces if you use double quotation marks before and after the name.

An IPX protocol and IPX network VLAN cannot both be configured to operate simultaneously on a Brocade switch or router. This restriction is also true for IP and IP sub-net VLANs.

Configuring a NetBIOS Protocol VLAN

You can create a NetBIOS protocol VLAN. The name appears in VLAN show displays. All ports of the system are assumed, by default, to be members of the VLAN when initially created. VLAN Membership can be modified using the **dynamic**, **static**, or **exclude** commands.

To create a NetBIOS Protocol VLAN on an 18 port device with permanent port membership of 4 and 5 and ports 8 through 12 as dynamic member ports, enter commands such as the following:

```
ServerIron(config)#netbios-proto
ServerIron(config-netbios-proto)#static e4 e5
ServerIron(config-netbios-proto)#exclude e1 to 3 e6 e7 e13 to 18
```

Syntax: [no] netbios-proto [<name>]

 © 2012 Brocade Communications Systems, Inc.

The <name> can be up to 16 characters long and can contain blanks

Configuring Another Protocol VLAN

You can create another protocol VLAN on the system. All ports of the switch are by default dynamically assigned to the newly created VLAN. VLAN Membership can be modified using the **dynamic**, **static**, or **exclude** commands.

You can use this option to define a protocol-based VLAN for protocols that are not specified as supported protocol VLANs on a switch or router, or do not require dedicated, separate broadcast domains.

On a 16 port ServerIron, ports 13 through 16 represent protocols Decnet and AppleTalk. You do not need to separate traffic by protocol into separate broadcast domains. Instead, create an Other Protocol VLAN with just those ports as members:

```
ServerIron(config)#other-proto
ServerIron(config-other-proto)#static e13 to 16
ServerIron(config-other-proto)#exclude e1 to 12
ServerIron(config-other-proto)#exit
```

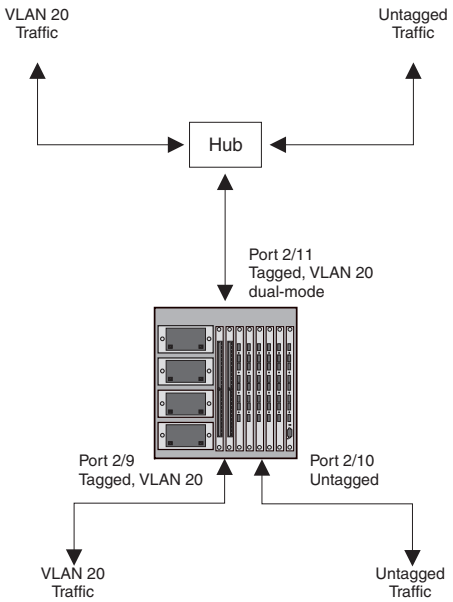
Syntax: [no] other-proto [<name>]

Dual-Mode VLAN Ports

Configuring a tagged port as ***dual-mode*** allows it to accept and transmit both tagged and untagged traffic simultaneously. A dual-mode port accepts and transmits frames belonging to VLANs configured for the port, as well as frames belonging to the default VLAN (untagged traffic).

For example, in Figure 2.1, port 2/11 is a dual-mode port belonging to VLAN 20. Traffic for VLAN 20, as well as traffic for the default VLAN, flows from a hub to this port. The dual-mode feature allows traffic for VLAN 20 and untagged traffic to go through the port at the same time.

Figure 2.1 Dual-mode VLAN port example



To enable the dual-mode feature on port 2/11 in Figure 2.1 enter the following commands:

```
ServerIron(config)#vlan 20
ServerIron(config-vlan-20)#tagged e 2/11
ServerIron(config-vlan-20)#tagged e 2/9
ServerIron(config-vlan-20)#int e 2/11
```

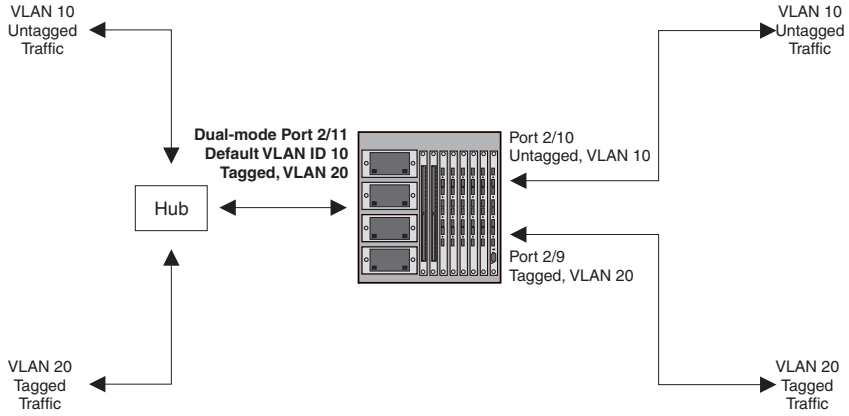
 © 2012 Brocade Communications Systems, Inc.

```
ServerIron(config-if-e100-2/11)#dual-mode
ServerIron(config-if-e100-2/11)#exit
```

Syntax: [no] dual-mode

Starting with Release 09.5.02a, you can configure a dual-mode port to transmit traffic for a specified VLAN (which is defined as Default VLAN) as untagged, while transmitting traffic for other VLANs as tagged. Figure 2.2 illustrates this enhancement.

Figure 2.2 Specifying a default VLAN ID for a dual-mode port



In Figure 2.2, tagged port 2/11 is a dual-mode port belonging to VLANs 10 and 20. The default VLAN assigned to this dual-mode port is 10. This means that the port transmits tagged traffic on VLAN 20 (and all other VLANs to which the port belongs) and transmits untagged traffic on VLAN 10.

The dual-mode feature allows tagged traffic for VLAN 20 and untagged traffic for VLAN 10 to go through port 2/11 at the same time. A dual-mode port transmits only untagged traffic on its default VLAN (that is, either VLAN 1, or a user-specified VLAN ID), and only tagged traffic on all other VLANs.

The following commands configure VLANs 10 and 20 in Figure 2.2. Tagged port 2/11 is added to VLANs 10 and 20, then designated a dual-mode port whose specified default VLAN is 10. In this configuration, port 2/11 transmits only untagged traffic on VLAN 10 and only tagged traffic on VLAN 20.

```
ServerIron(config)#vlan 10 by port
ServerIron(config-vlan-10)#untagged e 2/10
ServerIron(config-vlan-10)#tagged e 2/11
ServerIron(config-vlan-10)#exit

ServerIron(config)#vlan 20 by port
ServerIron(config-vlan-20)#tagged e 2/9
ServerIron(config-vlan-20)#tagged e 2/11
ServerIron(config-vlan-20)#exit

ServerIron(config)#int e 2/11
ServerIron(config-if-e100-2/11)#dual-mode 10
ServerIron(config-if-e100-2/11)#exit
```

Syntax: [no] dual-mode [<vlan-id>]

Notes:

- If you do not specify a <vlan-id> in the **dual mode** command, the port's default VLAN is set to 1. The port transmits untagged traffic on the DEFAULT-VLAN.
- The dual-mode feature is disabled by default. Only tagged ports can be configured as dual-mode ports.
- In trunk group, either all of the ports must be dual-mode, or none of them can be.

 © 2012 Brocade Communications Systems, Inc.

The **show vlan** command displays a separate row for dual-mode ports on each VLAN. For example:

```
ServerIron(config)#show vlan
Total PORT-VLAN entries: 3
Maximum PORT-VLAN entries: 16

legend: [S=Slot]

PORT-VLAN 1, Name DEFAULT-VLAN, Priority level0, Spanning tree Off
 Untagged Ports: (S1)  1  2  3  4  5  6  7  8
 Untagged Ports: (S2)  1  2  3  4  5  6  7  8 12 13 14 15 16 17 18 19
 Untagged Ports: (S2) 20 21 22 23 24
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: None
PORT-VLAN 10, Name [None], Priority level0, Spanning tree Off
 Untagged Ports: (S2) 10
   Tagged Ports: None
   Uplink Ports: None
 DualMode Ports: (S2) 11
PORT-VLAN 20, Name [None], Priority level0, Spanning tree Off
 Untagged Ports: None
   Tagged Ports: (S2)  9
   Uplink Ports: None
 DualMode Ports: (S2) 11
```

MAC Filters

The following sections describe how to configure MAC filters for Layer 2 operations.

Configuring Filters for Layer 2 Filtering

A MAC filter enables you to explicitly permit or deny switching of a Layer 2 packet received by the Brocade device. When the device receives a Layer 2 packet for switching, the device checks the packet's contents against the defined MAC filters. If the packet matches a filter, the system takes the action specified in the filter.

- If the action is permit, the system allows the packet to be switched.
- If the action is deny, the system immediately drops the packet.

To ensure security, if a packet does not match any of the MAC filters defined on the system, the system drops the packet by default. To configure the system to permit packets by default, you must define the last MAC filter in the filter list to allow all packets.

MAC filters can evaluate packets based on criteria such as source address and mask, destination address and mask, and protocol type (IP, ARP, and so on).

NOTE: You cannot use Layer 2 filters to filter Layer 4 information. To filter Layer 4 information, use IP filters.

NOTE: You cannot use Layer 2 filters to filter Layer 4 information. To filter Layer 4 information, use ACLs.The standard and extended ACLs described in that chapter are supported on the ServerIron.

To define filters for Layer 2 filtering on MAC addresses, enter commands such as the following:

```
ServerIron(config)#mac filter 1 deny 3565.3475.3676 ffff.0000.0000 any etype eq 806
ServerIron(config)#mac filter 1024 permit any any
ServerIron(config)#int e 1/1
```

 © 2012 Brocade Communications Systems, Inc.

```
ServerIron(config-if-1/1)#mac filter-group 1
```

These commands configure a filter to deny ARP traffic with a source MAC address that begins with “3565” to any destination. The second filter permits all traffic that is not denied by another filter.

After you define the filters, you apply them to individual interfaces using the **mac filter-group** command.

Syntax: [no] mac filter <filter-num> permit | deny <src-mac> <mask> | any <dest-mac> <mask> | any etype | llc | snap eq | gt | lt | neq <frame-type>

The <filter-num> is 1 – 64 (64 is the default system-max setting). If you use the **system-max mac-filter-sys** command, you can increase the maximum number of MAC filters support to 128 for global filter definitions.

The **permit | deny** argument determines the action the software takes when a match occurs.

The <src-mac> <mask> | **any** parameter specifies the source MAC address. You can enter a specific address value and a comparison mask or the keyword any to filter on all MAC addresses. Specify the mask using f’s (ones) and zeros. For example, to match on the first two bytes of the address aabb.ccdd.eeff, use the mask ffff.0000.0000. In this case, the filter matches on all MAC addresses that contain "aabb" as the first two bytes. The filter accepts any value for the remaining bytes of the MAC address. If you specify **any**, do not specify a mask. In this case, the filter matches on all MAC addresses.

The <dest-mac> <mask> | **any** parameter specifies the destination MAC address. The syntax rules are the same as those for the <src-mac> <mask> | **any** parameter.

Use the **etype** | **llc** | **snap** argument if you want to filter on information beyond the source and destination address. The MAC filter allows for you to filter on the following encapsulation types:

- **etype** (Ethertype) – a two byte field indicating the protocol type of the frame. This can range from 0x0600 to 0xFFFF.
- **llc** (IEEE 802.3 LLC1 SSAP and DSAP) – a two byte sequence providing similar function as the EtherType but for an IEEE 802.3 frame.
- **snap** (IEEE 802.3 LLC1 SNAP) – a specific LLC1 type packet.

To determine which type of frame is used on your network, use a protocol analyzer. If byte 12 of an Ethernet packet is equal to or greater than 0600 (hex), it is an Ethernet framed packet. Any number below this indicates an IEEE 802.3 frame (byte 12 will now indicate the length of the data field). Some well-known Ethernet types are 0800 (TCP/IP), 0600 (XNS), and 8137 (Novell Netware). Refer to RFC 1042 for a complete listing of EtherTypes.

For IEEE 802.3 frame, you can further distinguish the SSAP and DSAP of LLC header. Some well-known SAPs include: FE (OSI), F0 (NetBIOS), 42 (Spanning Tree BPDU), and AA (SNAP). Usually the DSAP and SSAP are the same.

NOTE: You must type in both bytes, otherwise the software will fill the field, left justified with a 00. Refer to RFC 1042 for a complete listing of SAP numbers.

SNAP is defined as an IEEE 802.3 frame with the SSAP, DSAP, and control field set to AA, AA, and 03. Immediately following these is a five-byte SNAP header. The first three bytes in this header are not used by the MAC filters. However, the next two bytes usually are set to the EtherType, so you can define the EtherType inside the SNAP header that you want to filter on.

The **eq** | **gt** | **lt** | **neq** argument specifies the possible operator: eq (equal), gt (greater than), lt (less than) and neq (not equal).

The <frame-type> argument is a hexadecimal number for the frame type. For example, the hex number for ARP is 806.

The **mac filter-group** <filter-list> applies a group of MAC filters to the interface. The filters must be applied as a group. For example, if you want to apply four filters to an interface, they must all appear on the same command line. You cannot add or remove individual filters in the group. To add or remove a filter on an interface, apply the filter group again containing all the filters you want to apply to the port. If you apply a filter group to a port that already has a filter group applied, the older filter group is replaced by the new filter group.

 © 2012 Brocade Communications Systems, Inc.

NOTE: Once you define a MAC filter, the device drops Layer 2 traffic that does not match a MAC permit filter.

Additional Examples of Layer 2 MAC Filter Definitions

```
ServerIron(config)#mac filter 1 permit any any etype eq 0800
```

This filter configures the device to permit (forward) any inbound packet with the Ethertype field set to 0800 (IP).

```
ServerIron(config)#mac filter 2 deny 0080.0020.000 ffff.ffff.0000 any etype eq 0800
```

This filter configures the device to deny an inbound packet with the first four bytes set to 0800.0020.xxxx and an EtherType field set to 0800 (IP). The destination field does not matter.

```
ServerIron(config)#mac filter 3 deny any 00e0.5200.1234 ffff.ffff.ffff snap eq 0800
```

This filter configures the device to deny any inbound IEEE 802.3 packet with a destination set to 00e0.5200.1234 and a SNAP EtherType set to 0800. The source address does not matter.

```
ServerIron(config)#mac filter 32 permit any any
```

This filter permits all packets. This filter is used as the last filter assigned in a filter-group that has previous deny filters in the group.

Abbreviating the Address or Mask

Address and Mask abbreviations are allowed. However, be careful when configuring them. The default fill character is a 0 and it will fill a byte range as left justified. This applies only to the MAC address and mask. A range of frame types cannot be filtered. Each frame type must be entered. Here are some examples.

```
ServerIron(config)#mac filter 1 deny 0800.0700 ffff.ff00 any
```

This command expands to the following: `mac filter 1 deny 0800.0700.0000 ffff.ff00.0000`. The filter shown above denied forwarding of an inbound frame that has the source address set to 080007 as the first three bytes. All other information is not significant.

Here is another example of the fill feature:

```
ServerIron(config)#mac filter 2 deny 0260.8C00.0102 0.0.ffff any
```

This command expands to the following: `mac filter 1 deny 0260.8C00.0102 0000.0000.ffff any`

Since the fill character is 0's and the fill is left justified, certain filters will not allow for abbreviations. For example, suppose you want to deny an inbound packet that contained a broadcast destination address. Enter the following command:

```
ServerIron(config)#mac filter 5 deny any ff ff
```

This command contains a destination of address all F's and mask of F's. The command expands to the following:

```
ServerIron(config)#mac filter 1 deny any 00ff.0000.0000 00ff.0000.0000
```

Here is another example for DSAP and SSAP:

```
ServerIron(config)#mac filter 10 deny any any llc eq F0
```

This command expands to the following: `mac filter 2 deny any any llc eq 00f0`

If you want to filter on both the SSAP and DSAP, then the following example shows this:

```
ServerIron(config)#mac filter 4 deny any 0020.0010.1000 ffff.ffff.0000 llc eq e0e0
```

Setting the MAC Age Time

To set the aging period for all address entries in the switch or router address table, enter the following command:

```
ServerIron(config)#mac-age 600
```

Syntax: [no] mac-age-time <value>

The <value> is 0 – 65535 seconds. The default is 300 seconds. If you specify 0, the entries do not age.

 © 2012 Brocade Communications Systems, Inc.

Enabling Logging of Packets Denied by Layer 2 MAC Filters

When you enable this feature, the device generates Syslog entries and SNMP traps for denied packets.

To enable logging of packets that are denied by Layer 2 MAC filters, enter the following command:

```
ServerIron(config)#mac filter log-enable
```

Syntax: [no] mac filter log-enable

Address-Lock Filters

An ***address-lock filter*** restricts the number of MAC addresses that a switch can learn from a specific port. After the switch learns the specified number of MAC addresses from the port, the switch stops learning addresses received on that port. In addition, the switch does not accept or forward traffic on the port unless the traffic contains one of the source or destination MAC addresses locked for the port.

Address-lock filters apply only to Layer 2 traffic and do not affect Layer 3 or Layer 4 traffic on the locked ports.

Unlike addresses learned from other ports, addresses learned from a locked port are not subject to aging.

Configuring a Broadcast Filter

You can filter on all broadcast traffic or on IP UDP broadcast traffic.

To configure a Layer 2 broadcast filter to filter all types of broadcasts, then apply the filter to ports 1, 2, and 3, enter commands such as the following:

```
ServerIron(config)# broadcast filter 1 any
ServerIron(config-bcast-filter-id-1)# exclude-ports ethernet 1 to 3
ServerIron(config-bcast-filter-id-1)# write mem
```

To configure two filters, one to filter IP UDP traffic on ports 1 – 4, and the other to filter all broadcast traffic on port 6, enter commands such as the following:

```
ServerIron(config)# broadcast filter 1 ip udp
ServerIron(config-bcast-filter-id-1)# exclude-ports ethernet 1 to 4
ServerIron(config-bcast-filter-id-1)# exit
ServerIron(config)# broadcast filter 2 any
ServerIron(config-bcast-filter-id-2)# exclude-ports ethernet 6
ServerIron(config-bcast-filter-id-2)# write mem
```

To configure an IP UDP broadcast filter and apply that applies only to port-based VLAN 10, then apply the filter to two ports within the VLAN, enter commands such as the following:

```
ServerIron(config)# broadcast filter 4 ip udp vlan 10
ServerIron(config-bcast-filter-id-4)# exclude-ports eth 1 eth 3
ServerIron(config-bcast-filter-id-1)# write mem
```

Syntax: [no] broadcast filter <filter-id> any | ip udp [vlan <vlan-id>]

The <filter-id> specifies the filter number and can be a number from 1 – 8. The software applies the filters in ascending numerical order. As soon as a match is found, the software takes the action specified by the filter (block the broadcast) and does not compare the packet against additional broadcast filters.

You can specify **any** or **ip udp** as the type of broadcast traffic to filter. The **any** parameter prevents all broadcast traffic from being sent on the specified ports. The **ip udp** parameter prevents all IP UDP broadcasts from being sent on the specified ports but allows other types of broadcast traffic.

If you specify a port-based VLAN ID, the filter applies only to the broadcast domain of the specified VLAN, not to all broadcast domains (VLANs) on the device.

As soon as you press Enter after entering the command, the CLI changes to the configuration level for the filter you are configuring. You specify the ports to which the filter applies at the filter's configuration level.

 © 2012 Brocade Communications Systems, Inc.

Syntax: [no] exclude-ports ethernet <portnum> to <portnum>

Or

Syntax: [no] exclude-ports ethernet <portnum> ethernet <portnum>

These commands specify the ports to which the filter applies.

NOTE: This is the same command syntax as that used for configuring port-based VLANs. Use the first command for adding a range of ports. Use the second command for adding separate ports (not in a range). You also can combine the syntax. For example, you can enter **exclude-ports ethernet 1/4 ethernet 2/6 to 2/9**.

Setting the Broadcast Limit

You can specify the maximum number of broadcast packets the device can forward each second. By default the device sends broadcasts and all other traffic at wire speed and is limited only by the capacities of the hardware. However, if other devices in the network cannot handle unlimited broadcast traffic, this command allows you to relieve those devices by throttling the broadcasts at the device.

The broadcast limit does not affect multicast or unicast traffic. However, you can use the **multicast limit** and **unknown-unicast limit** <limit> global commands to control these types of traffic. The **unknown-unicast limit** command specifies the maximum number of unknown-unicast packets the device can forward each second. By default the device sends unknown unicasts and all other traffic at wire speed and is limited only by the capacities of the hardware. However, if other devices in the network cannot handle unlimited unknown-unicast traffic, this command allows you to relieve those devices by throttling the unknown unicasts at the Brocade device.

For example, enter the following command:

```
ServerIron(config)#broadcast limit 30000
```

To enable an interface specific configuration, enter commands such as the following:

```
ServerIron(config)#int e 6
ServerIron(config-if-6)#broadcast limit 30000
```

Syntax: [no] broadcast limit <num>

Assigning a Gateway List

Dynamic Host Configuration Protocol (DHCP) Assist allows a Brocade switch to assist a router that is performing multinetting on its interfaces as part of its DHCP relay function. DHCP eliminates the need to manually assign IP addresses to clients. Instead of each client having a statically configured IP address, clients petition a server for IP addresses when the clients are booted.

DHCP Assist ensures that a DHCP server that manages multiple IP subnets can readily recognize the requester's IP sub-net, even when that server is not on the client's local LAN segment. The Brocade switch does this by stamping the correct gateway IP address into a DHCP discovery packet on behalf of the router.

Use the **dhcp-gateway-list** <num> <ip-addr> command when DHCP Assist is enabled on a Brocade switch. A gateway address must be defined for each sub-net that will be requesting addresses from a DHCP server. This allows the stamping process to occur. Each gateway address defined on the switch corresponds to an IP address of the ServerIron interface or other device involved.

Up to eight addresses can be defined for each gateway list in support of ports that are multi-homed. When multiple IP addresses are configured for a gateway list, the switch inserts the addresses into the discovery packet in a round robin fashion.

Up to 32 gateway lists can be defined for each switch.

For example, enter the following command:

```
ServerIron(config)#dhcp-gateway-list 1 192.95.5.1
```

 © 2012 Brocade Communications Systems, Inc.

Or, assign it to a specific interface:

```
ServerIron(config)#int e 2
ServerIron(config-if-2)#dhcp-gateway-list 1
```

Syntax: [no] dhcp-gateway-list <num> <ip-addr>

Multicast

Enabling IP Multicast Traffic Reduction

IP multicast containment allows Brocade switches to limit switching of IP multicast packets to only those ports on the switch that are identified as IP multicast members. Brocade switches can provide IP multicast containment in either of the following modes:

- **Passive**—The switch listens for Internet Group Membership Protocol (IGMP) packets and forwards them to the appropriate ports.
- **Active**—The switch actively sends out host queries to identify IP multicast groups on the network and inserts this information into the IGMP packets.

Routers in the network generally handle host queries. Unless your configuration does not contain a router to provide this service, use IP multicast containment in the passive mode.

The ServerIron can operate in either an **active** or **passive** IP multicast mode. You must save changes to flash and reset (**reload**) the switch for the configuration changes to become active.

If configured to be **active**, the switch will actively send out host queries to identify IP Multicast groups on the network and insert this information in the IGMP packet. Routers in the network generally handle this operation

If configured to be **passive**, the switch will only identify the packet as an IGMP packet and forward it accordingly.

To enable IP Multicast Traffic Reduction, enter commands such as the following:

```
ServerIron(config)#ip multicast passive
ServerIron(config)#write memory
ServerIron(config)#end
ServerIron#reload
```

Syntax: [no] ip multicast active | passive

Configuring a Multicast Filter

You can filter on all multicast packets or on specific multicast groups.

To configure a Layer 2 multicast filter to filter all multicast groups, then apply the filter to ports 2/4, 2/5, and 2/8, enter commands such as the following:

```
ServerIron(config)#multicast filter 1 any
ServerIron(config-mcast-filter-id-1)#exclude-ports e 2/4 to 2/5 e 2/8
ServerIron(config-mcast-filter-id-1)#write mem
```

To configure a multicast filter to block all multicast traffic destined for multicast addresses 0100.5e00.5200 – 0100.5e00.52ff on port 4/8, enter commands such as the following:

```
ServerIron(config)#multicast filter 2 any 0100.5e00.5200 ffff.ffff.ff00
ServerIron(config-mcast-filter-id-2)#exclude-ports ethernet 4/8
ServerIron(config-mcast-filter-id-2)#write mem
```

The software calculates the range by combining the mask with the multicast address. In this example, all but the last two bits in the mask are “significant bits” (ones). The last two bits are zeros and thus match on any value.

Syntax: [no] multicast filter <filter-id> any | ip udp mac <multicast-address> | any [mask <ip-mask>] [vlan <vlan-id>]

 © 2012 Brocade Communications Systems, Inc.

The parameter values are the same as the for the broadcast filter command. In addition, the multicast filter command requires the **mac** <multicast-address> I **any** parameter, which specifies the multicast address. Enter **mac any** to filter on all multicast addresses. Enter **mac** followed by a specific multicast address to filter only on that multicast address.

To filter on a range of multicast addresses, use the **mask** <ip-mask> parameter. For example, to filter on multicast groups 0100.5e00.5200 – 0100.5e00.52ff, use mask ffff.ffff.ff00. The default mask matches all bits (is all Fs). You can leave the mask off if you want the filter to match on all bits in the multicast address.

Setting the Multicast Limit

By default, the device sends multicasts and all other traffic at wire speed and is limited only by the capacities of the hardware. However, if other devices in the network cannot handle unlimited multicast traffic, this command allows you to relieve those devices by throttling the multicasts at the Brocade device.

NOTE: The multicast limit does not affect broadcast or unicast traffic. However, you can use the **broadcast limit** and **unknown-unicast limit** commands to control these types of traffic.

To specify the maximum number of multicast packets the device can forward each second, enter the following command:

```
ServerIron(config)#multicast limit 30000
```

To put the multicast limit on a specific interface, enter commands such as the following:

```
ServerIron(config)#interface e5
ServerIron(config-if-5)#multicast limit 30000
```

Syntax: [no] multicast limit <num>

<num> can be 0 – 4294967295.

Disabling IGMP Queries

You can disable Internet Group Membership Protocol (IGMP) queries from being sent or received on the port. IGMP queries are enabled by default.

To disable IGMP queries on an interface, enter commands such as the following:

```
ServerIron(config)#int e5
ServerIron(config-if-5)#ip-multicast-disable
```

To re-enable the IGMP queries on the interface, enter a command such as the following:

```
ServerIron(config-if-5)#no ip-multicast-disable
```

Syntax: [no] ip-multicast-disable

Notes:

- This feature is supported only in hot-standby configurations.
- IP forwarding must be enabled on the ServerIron
- The backup MAC address in the hot-standby configuration must be the first port MAC address of one of the ServerIrons.
- All VIPs on the device must be in a VE subnet.
- This feature is not supported in SSLB, TCS, FWLB, or IP NAT configurations.

Adding an IP Interface

To add an IP interface to the ServerIron, you must first add a virtual routing interface and then configure IP addresses on it.

 © 2012 Brocade Communications Systems, Inc.

To add a virtual routing interface, enter commands such as the following:

```
ServerIron(config)# vlan 1
ServerIron(config-vlan-1)# router-interface ve 1
```

The **vlan 1** command changes the CLI to the configuration level for VLAN 1. The **router-interface ve 1** command adds virtual routing interface 1.

Syntax: [no] router-interface ve <num>

The <num> parameter specifies the interface ID and can be from 1 – 24.

After you add a virtual routing interface, you can add up to 64 IP interfaces to the virtual routing interface. To add an IP interface, use the following CLI method.

NOTE: When you add an IP interface to a virtual routing interface and the interface is up, the software adds a directly-connected static IP route to the route table for the address' sub-net. The software does not add the route unless the interface is up.

NOTE: For IP forwarding to work properly, you must add an IP interface that is in the same sub-net as the management IP address. This is true regardless of whether you plan to allow management access from other sub-nets.

NOTE: Do not configure a virtual routing interface to have the same IP address as the ServerIron's management address.

To add an IP interface, enter commands such as the following:

```
ServerIron(config)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
```

The **interface ve 1** command changes the CLI to the configuration level for virtual routing interface 1. The **ip address** command adds an IP interface.

Syntax: [no] ip address | nat-address | standby-address <ip-addr> <ip-mask> I <ip-addr>/<mask-bits>

The **address | nat-address | standby-address** parameter identifies the type of IP interface you are adding.

- The **address** parameter adds a standard IP interface. This option is applicable in most cases.
- The **nat-address** parameter applies to active-standby configurations. This parameter configures a shared IP interface for use with SLB source NAT. Enter the same command with the same IP address on each of the ServerIron's in the active-standby configuration. The address is active only on one ServerIron' (the ServerIron' that is currently active) at a time.
- The **standby-address** parameter applies to active-standby configurations and allows both ServerIrons to share the same router interface. One of the ServerIrons actively supports the interface while the other ServerIron provides failover for the interface if the first ServerIron becomes unavailable. Real servers can use the shared interface as their default gateway. Enter the same command with the same IP address on each of the ServerIrons in the active-standby configuration. The address is active only on one ServerIron (the ServerIron that is currently active) at a time.

The <ip-addr> parameter specifies the IP address.

The <ip-mask> parameter specifies a class-based (or "Classical") IP sub-net mask.

The <mask-bits> parameter specifies the number of significant bits in a Classless Interdomain Routing (CIDR) sub-net mask.

You can use either format to configure the interface. For example, both the following commands are valid and produce the same result:

- ip address 10.10.10.1 255.255.255.0
- ip address 10.10.10.1/24

 © 2012 Brocade Communications Systems, Inc.

Adding a Static IP Route

The software places the static route in the IP route table only if the virtual routing interface is up.

To add a static IP route to the 209.157.2.x/24 sub-net, enter a command such as the following:

```
ServerIron(config)#ip route 209.157.2.0 255.255.255.0 192.168.2.1
```

Syntax: [no] ip route <dest-ip-addr> <dest-mask> <next-hop-ip-addr> | null0 [<metric>]

Syntax: [no] ip route <dest-ip-addr>/<mask-bits> <next-hop-ip-addr> | null0 [<metric>]

The <dest-ip-addr> parameter specifies the route's destination. The <dest-mask> is the network mask for the route's destination IP address. Alternatively, you can specify the network mask information by entering a forward slash followed by the number of bits in the network mask. For example, you can enter 192.0.0.0 255.255.255.0 as 192.0.0.0/.24. To configure a default route, enter 0.0.0.0 for <dest-ip-addr> and 0.0.0.0 for <dest-mask> (or 0 for the <mask-bits> if you specify the address in CIDR format). Specify the IP address of the default gateway using the <next-hop-ip-addr> parameter.

The <next-hop-ip-addr> parameter specifies the IP address of the next-hop router (gateway) for the route. If you specify null0 instead of a next hop IP address, the ServerIron discards packets addressed to the route's destination IP address instead of forwarding them to another device.

If you add a default route, the gateway address of the route replaces the default gateway address configured by the ip default-gateway command. Likewise, if you use the ip default-gateway command to change the default gateway address, the gateway address in the default route is automatically changed also.

The <metric> parameter specifies the cost of the route and can be a number from 1 – 16. The default is 1. The metric is used by RIP. If you do not enable RIP, the metric is not used.

NOTE: The ServerIron supports IP fragmentation but is not in the position to perform IP fragmentation. It is recommended to perform IP fragmentation on routers connecting to the ServerIron.

Adding a Static ARP Entry

Static entries are useful in cases where you want to pre-configure an entry for a device that is not connected to the ServerIron, or you want to prevent a particular entry from aging out. The software removes a dynamic entry from the ARP cache if the ARP aging interval expires before the entry is refreshed. Static entries do not age out, regardless of whether the ServerIron receives an ARP request from the device that has the entry's address. The software places a static ARP entry into the ARP cache as soon as you create the entry.

To add a static ARP entry, enter commands such as the following:

```
ServerIron(config)#arp 1 209.157.22.3 aaaa.bbbb.cccc ethernet 3
```

This command adds a static ARP entry that maps IP address 209.157.22.3 to MAC address aaaa.bbbb.cccc. The entry is for a MAC address connected to ServerIron port 3.

Syntax: [no] arp <num> <ip-addr> <mac-addr> ethernet <portnum> [vlan <vlan-id>]

The <num> parameter specifies the entry number. You can specify a number from 1 up to the maximum number of static entries allowed on the device. You can allocate more memory to increase this amount to 128 entries. See "Setting System Max" on page 2-23.

The <ip-addr> parameter specifies the IP address of the device that has the MAC address of the entry.

The <mac-addr> parameter specifies the MAC address of the entry.

The ethernet <portnum> parameter specifies the port number attached to the device that has the MAC address of the entry.

The vlan <vlan-id> parameter specifies the port-based VLAN the entry belongs to. This parameter is required if the port you specify is a member of more than one port-based VLAN. Otherwise, the parameter is optional.

To display the static ARP entries, see "Displaying Static ARP Entries" on page 2-28.

 © 2012 Brocade Communications Systems, Inc.

NOTE: You can add static ARP entries regardless of whether IP forwarding is enabled. On software release 08.x.xxR, you must create the static MAC that corresponds to the static ARP before creating a static ARP entry.

Clearing the ARP Cache

To remove all data from the ARP cache, enter the following commad:

```
ServerIron# clear arp
```

To clear all ARP entries for port 2 on the module in slot 3, enter commands such as the following:

```
ServerIron# clear arp ethernet 3/2
```

Syntax: clear arp [ethernet <num> | mac-address <xxxx.xxxx.xxxx> [<mask>] | <ip-addr> [<ip-mask>]]

Specify the MAC address mask as “f”s and “0”s, where “f”s are significant bits. Specify IP address masks in standard decimal mask format (for example, 255.255.0.0).

NOTE: The **clear arp** command clears learned ARP entries but does not remove any static ARP entries.

Clearing the IP Cache

To remove all entries from the IP cache, enter the following command:

```
ServerIron# clear ip cache
```

Syntax: clear ip cache

Clearing the MAC Address Table

To remove all entries in the MAC address table, enter the following commad:

```
ServerIron#clear mac-address
```

Syntax: clear mac-address

Setting System Max

Use **system-max** <option> to modify the default settings for parameters that use system memory. The configurable parameters and their defaults and maximums differ depending on the device.

Issue the **show default values** command to display the configurable parameters, their defaults. If you specify default but not the optional values, the default states for parameters that can either be enabled or disabled are displayed. If you also specify values, the default values for parameters that take a numeric value are displayed.

To increase the number of real servers available on the ServerIron, enter commands such as the following:

```
ServerIron(config)#system-max l4-real 2048
```

To increase the number of virtual servers available on the ServerIron, enter commands such as the following:

```
ServerIron(config)#system-max l4-virtual-server 512
```

To increase the number of TCP/UDP ports available on the ServerIron, enter a command such as the following:

```
ServerIron(config)#system-max l4-server-port 4095
```

To increase the number of TCP buffers available on the ServerIron, enter a command such as the following:

```
ServerIron(config)#system-max tcp-buffer 2048
```

To increase the number of SNMP views available on the ServerIron, enter a command such as the following:

 © 2012 Brocade Communications Systems, Inc.

```
ServerIron(config)#system-max view 15
```

Syntax: [no] system-max <option>

<option> can be any of the following:

- l4-real-server <real-servers>—where <real-servers> can be from 64 – 2048.
- l4-virtual-server <virtual-servers>—where <virtual-servers> can be from 64 – 512.
- l4-server-port <number-of-ports>—where <number-of-ports> can be from 256 – 4096. The system max of 4096 includes the default port for each defined real server.
- tcp-buffer <number-of-buffers>—where <number-of-buffers> can be from 128 – 2048. The ServerIron uses TCP buffers for TCP sessions. Applications such as GSLB use many TCP buffers, since buffers are required for TCP health checks as well as client connections with real servers. If you receive a message that the ServerIron cannot perform a health check or other TCP tasks, you might need to allocate more memory for TCP buffers.
- <number-of-views>—specifies the maximum number of SNMPv2 and v3 views that can be configured on a ServerIron. The number of views can be from 10 – 65536. The default is 10 views.

Adding a Static MAC Address

To define a static MAC addresses on an individual switch or switching port to ensure it is not aged out , enter commands such as the following:

```
ServerIron(config)#static-mac-address 1145.5563.67FF e12 7 router-type
```

The syntax for adding static MAC entries differs depending on whether you are using a stackable or chassis ServerIron. To create a static MAC entry that is associated with multiple ports:

```
ServerIron(config)#static-mac-address aaaa.bbbb.cccc ethernet 1 ethernet 3 to 5
```

This command creates a static MAC entry that is associated with port 1 and ports 3 – 5. The ServerIron forwards traffic addressed to aaaa.bbbb.cccc out all the ports you specify, in this case 1, 3, 4, and 5.

If you enter the command at the global CONFIG level, the static MAC entry applies to the default port-based VLAN (VLAN 1). If you enter the command at the configuration level for a specific port-based VLAN, the entry applies to that VLAN and not to the default VLAN.

If you want to include a trunk group when you configure a static MAC entry that has multiple ports, include only the primary port of the trunk group. If you include all the trunk group's ports, the ServerIron uses all the ports to forward traffic for the MAC address instead of using only the active trunk port.

To enter a static MAC address entry for port 5, that is also resident in port-based VLAN 4, enter commands such as the following:

```
ServerIron(config)#vlan 4
ServerIron(config-vlan-4)#static-mac-address 023.876.735 ethernet 5 high-priority
router-type
```

To create a static ARP entry for a static MAC entry, enter commands such as the following:

```
ServerIron(config)#arp 1 192.53.4.2 aaaa.bbbb.cccc ethernet 1
```

The **arp** command allows you to specify only one port number. To create a static ARP entry for a static MAC entry that is associated with multiple ports, specify the first (lowest-numbered) port associated with the static MAC entry.

Syntax: [no[static-mac-address <mac-addr> ethernet <portnum> [priority <0-7>] [host-type | router-type]

The priority can be 0 – 7 (0 is lowest and 7 is highest). The default is host-type.

Brocade recommends that you configure a static ARP entry to match the static MAC entry. In fact, the software automatically creates a static MAC entry when you create a static ARP entry. When a static MAC entry has a corresponding static ARP entry, you cannot delete the static MAC entry unless you first delete the static ARP entry.

 © 2012 Brocade Communications Systems, Inc.

Displaying IP Forwarding Information

You can display the following IP forwarding information:

- The IP forwarding state (enabled or disabled)
- ARP entries
- IP interfaces
- The IP route table
- IP traffic statistics

Displaying IP Forwarding State Information

To display IP forwarding state information as well as other global IP parameters, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip

             Enabled : IP_Forwarding

             Disabled : RIP  RIP-Redist

     Switch IP address: 192.168.2.100

           Subnet mask: 255.255.255.0

Default router address: 192.168.2.1
   TFTP server address: None
Configuration filename: None
        Image filename: None
```

Syntax: show ip

This display shows the following information.

Table 2.1: CLI Display of Global IP Configuration Information

This Field...	Displays...
IP configuration	
IP Forwarding state	The state of the IP forwarding feature. The state can be one of the following: • Disabled • Enabled
RIP state	The state of RIP. The state can be one of the following: • Disabled • Enabled If route redistribution is enabled, “RIP -Redist” is displayed as well. For information, see “Redistributing IP Static Routes into RIP” on page 2-37.

 © 2012 Brocade Communications Systems, Inc.

Table 2.1: CLI Display of Global IP Configuration Information (Continued)

This Field...	Displays...
Switch IP address	The management IP address you configured on the ServerIron. Specify this address for Telnet or Web management access.
Subnet mask	The sub-net mask for the management IP address.
Default router address	The address of the default gateway, if you specified one. **Note**: When IP forwarding is enabled, the address is listed only if the corresponding virtual interface is up. When IP forwarding is disabled, the configured default gateway address is always displayed.
Most recent TFTP access	
TFTP server address	The IP address of the most-recently contacted TFTP server, if the ServerIron has contacted a TFTP server since the last time the software was reloaded or the ServerIron was rebooted.
Configuration filename	The name under which the ServerIron's startup-config file was uploaded or downloaded during the most recent TFTP access.
Image filename	The name of the ServerIron flash image (system software file) that was uploaded or downloaded during the most recent TFTP access.

Displaying the IP Host Table

To display the IP host table showing indexes to MAC addresses and the IP address of the next hop for ServerIrons configured to operate in a multinetted environment, enter the following command:

```
ServerIron#[ 1] sh ip cache
         IP              Mac        Port Age VlanId  Cam CamF   Hw FCnt

209.157.20.1        0000.0000.0000     6   0   3144    0    0    0    0
```

Syntax: show ip cache [<ip-addr> [<ip-addr>]]

Displaying the ARP Cache or the Static ARP Table

You can display the ARP cache or the static ARP table. The ARP table contains the static ARP entries, if any, you configured on the device. The ARP cache contains all the ARP entries, including static entries.

To display the ARP cache, enter the following command at any level of the CLI:

```
ServerIron(config)# show arp
         IP              Mac            Type        Port Age VlanId

10.10.10.10        00d0.0958.9b07      Static         9   0      1
192.168.2.14       0050.04bb.81fa      Static        15   0      1
192.168.2.1        00e0.5205.9056      Static        15   0      1
192.168.2.157      00e0.2972.2ab5      Dynamic       15   0      1
192.168.2.15       0010.5ad1.3701      Dynamic       15   0      1
192.168.2.77       00e0.5202.de72      Dynamic       15   0      1
Total Arp Entries : 6
```

 © 2012 Brocade Communications Systems, Inc.

Syntax: show arp [<ip-addr> [<ip-mask>] | ethernet <portnum> mac-address <xxxx.xxxx.xxxx> [<mask>]]

The <ip-addr> and <ip-mask> parameters let you restrict the display to entries for a specific IP address and network mask. Specify the IP address masks in standard decimal mask format (for example, 255.255.0.0).

NOTE: The <ip-mask> parameter and <mask> parameter perform different operations. The <ip-mask> parameter specifies the network mask for a specific IP address, whereas the <mask> parameter provides a filter for displaying multiple MAC addresses that have specific values in common.

Specify the MAC address mask as "f"s and "0"s, where "f"s are significant bits. Specify IP address masks in standard decimal mask format (for example, 255.255.0.0).

The **ethernet** <portnum> parameter lets you restrict the display to entries for a specific port.

The **mac-address** <xxxx.xxxx.xxxx> parameter lets you restrict the display to entries for a specific MAC address.

The <mask> parameter lets you specify a mask for the **mac-address** <xxxx.xxxx.xxxx> parameter, to display entries for multiple MAC addresses. Specify the MAC address mask as "f"s and "0"s, where "f"s are significant bits.

Here are some examples of how to use these commands.

The following command displays all ARP entries for MAC addresses that begin with "abcd":

```
ServerIron# show arp mac-address a.b.c.d ffff.0000.0000
```

The following command displays all IP address entries for IP addresses that begin with "209.157":

```
ServerIron# show arp 209.157.0.0 255.255.0.0
```

This **show arp** command displays the following information.

Table 2.2: CLI Display of ARP Cache

This Field...	Displays...
IP	The IP address of the device.
MAC	The MAC address of the device.
Type	The type, which can be one of the following: • Dynamic – The ServerIron learned the entry from an incoming packet. • Static – You added the entry to the ARP table.
Port	The port on which the entry was learned.
Age	The number of minutes the entry has remained unused. If this value reaches the ARP aging period, the entry is removed from the table. **Note**: Static entries do not age out.
VlanId	The port-based VLAN that the ServerIron port connected to the entry's MAC address is in.
Total ARP Entries	The total number of entries in the cache. The total includes both dynamic (learned) and static ARP entries.

 © 2012 Brocade Communications Systems, Inc.

Displaying Static ARP Entries

To display static ARP entries, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip static-arp
Static ARP table size: 64, configurable from 64 to 128
  Index   IP Address          MAC Address         Port
  1       10.10.10.10         00d0.0958.9b07      9
  2       192.168.2.1         00e0.5205.9056      15
  3       192.168.2.157       00e0.2972.2ab5      15
  4       192.168.2.14        0050.04bb.81fa      15
  5       192.168.2.15        0010.5ad1.3701      15
```

Syntax: show ip static-arp [<ip-addr> [<ip-mask>] | ethernet <portnum> mac-address <xxxx.xxxx.xxxx> [<mask>]]

The parameters are the same as those for the **show arp** command.

The **show ip static-arp** command displays the following information.

Table 2.3: CLI Display of Static ARP Table

This Field...	Displays...
Static ARP table size	The maximum number of static entries that can be configured on the device using the current memory allocation. The range of valid memory allocations for static ARP entries is listed after the current allocation. To change the memory allocation for static ARP entries, see “Setting System Max” on page 2-23.
Index	The number of this entry in the table. You specify the entry number when you create the entry.
IP Address	The IP address of the device.
MAC Address	The MAC address of the device.
Port	The port attached to the device the entry is for.

Displaying a List of IP Interfaces

To display a list of the IP interfaces configured on the ServerIron, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip interface
Interface       IP-Address      OK?  Method    Status                Protocol
Ve 1            192.168.2.1     YES  manual    up                    up
Ve 1            10.10.10.1      YES  manual    up                    up
Ve 1            20.20.20.1      YES  manual    up                    up
Ve 10           120.120.120.1   YES  manual    down                  up
Ve 10           130.130.130.1   YES  manual    down                  up
```

Syntax: show ip interface

 © 2012 Brocade Communications Systems, Inc.

This command displays the following information.

Table 2.4: CLI Display of IP Interfaces

This Field...	Displays...
Interface	The virtual routing interface.
IP-Address	The IP address of the interface.
OK?	Whether the IP address has been configured on the interface.
Method	Whether the IP address has been saved in NVRAM. If you have set the IP address for the interface in the CLI or Web Management interface, but have not saved the configuration, the entry for the interface in the Method field is “manual”.
Status	The link status of the interface. The status can be one of the following: • down • up
Protocol	Whether the interface can provide two-way communication. If the IP address is configured, and the link status of the interface is up, the entry in the protocol field is “up”. Otherwise the entry in the protocol field is “down”.

Displaying the IP Route Table

To display the IP route table, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip route
Total number of IP routes: 9
Start index: 1  D:Connected  S:Static *:Candidate default
      Destination       NetMask           Gateway           Port   Cost   Type
1     10.10.10.0        255.255.255.0     0.0.0.0           ve1    1      D
2     20.20.20.0        255.255.255.0     0.0.0.0           ve1    1      D
3     50.50.50.0        255.255.255.0     20.20.20.10       ve1    1      S
4     60.60.60.0        255.255.255.0     20.20.20.10       ve1    1      S
5     70.70.70.0        255.255.255.0     120.120.120.10    ve1    1      S
6     120.120.120.0     255.255.255.0     0.0.0.0           ve1    1      D
7     130.130.130.0     255.255.255.0     0.0.0.0           ve1    1      D
8     192.168.2.0       255.255.255.0     0.0.0.0           ve1    1      D
9     0.0.0.0           0.0.0.0           192.168.2.1       ve1    1      S
```

Syntax: show ip route

 © 2012 Brocade Communications Systems, Inc.

This command displays the following information.

Table 2.5: CLI Display of IP Route Table

This Field...	Displays...
Total number of IP routes	The total number of routes in the table, including routes that you added and directly-connected routes the software added when you added IP interfaces.
Start index	The starting entry number in the table.
Destination	The destination network of the route.
NetMask	The network mask of the destination address.
Gateway	The next-hop router.
Port	The virtual routing interface to which the route belongs.
Cost	The route's cost.
Type	The route type, which can be one of the following: • D – The destination is directly connected to the ServerIron. • R – The route is a RIP route. • S – The route is a static route.

Displaying IP Forwarding Traffic Statistics

To display IP forwarding traffic statistics, enter the following command at any level of the CLI:

```
ServerIron(config)# show ip traffic
IP Statistics
  587 received, 593 sent, 14 forwarded
  0 fragmented, 0 reassembled, 0 bad header
  489 no route, 0 unknown proto, 0 no buffer, 9 other errors
```

Syntax: show ip traffic

This command displays the following information related to IP forwarding.

Table 2.6: CLI Display of IP Forwarding Traffic Statistics

This Field...	Displays...
received	The total number of IP packets received by the device.
sent	The total number of IP packets originated and sent by the device.
forwarded	The total number of IP packets received by the device and forwarded to other devices.
filtered	The total number of IP packets filtered by the device.

 © 2012 Brocade Communications Systems, Inc.

Table 2.6: CLI Display of IP Forwarding Traffic Statistics (Continued)

This Field...	Displays...
fragmented	The total number of IP packets fragmented by this device to accommodate the MTU of this device or of another device.
reassembled	The total number of fragmented IP packets that this device re-assembled.
bad header	The number of IP packets dropped by the device due to a bad packet header.
no route	The number of packets dropped by the device because there was no route.
unknown proto	The number of packets dropped by the device because the value in the Protocol field of the packet header is unrecognized by this device.
no buffer	This information is used by Brocade customer support.
other errors	The number of packets that this device dropped due to error types other than the types listed above.

The display contains additional sections of statistics. However, the additional statistics apply to Layer 4 – 7 switching, not to IP forwarding.

Clearing IP Traffic Statistics

To clear the IP traffic statistics displayed by the **`show ip traffic`** command, enter the following command:

```
ServerIron# clear ip traffic
```

Syntax: clear ip traffic

IP Interfaces and Multinetting

Beginning with Release 8.0.00, the ServerIron Chassis devices support Layer 3 features, including the following:

- Multiple IP interfaces in the same or different sub-nets
- Support for multiple sub-net addresses on the same physical port or a single sub-net address on multiple physical ports
- Route-only support

NOTE: When switch trunk is configured and the servers are all remote servers 1 hop away from the ServerIron, Layer 3 switch trunking does not work. The ServerIron does not perform load balancing on the trunk ports. The expected behavior is trunking should be based on destination IP only.

The Layer 3 features include support for configuring multiple IP interfaces in the same or different IP sub-nets on the ServerIron.

- Without Layer 3 – When you use the ServerIron as a Layer 2 and Layer 4 – 7 switch, you can configure one IP interface on the device. The address is used as the management interface for the device. To multi-net the device, you must configure source IP addresses for the sub-nets that are in addition to the sub-net containing the device's management IP address.
- With Layer 3 – When you use the ServerIron as a Layer 2/3 and Layer 4 – 7 switch, you can configure separate IP sub-net interfaces on individual ports. In addition, you can associate the same sub-net interface with all the ports in a port-based VLAN by configuring a virtual routing interface on the VLAN, then assigning an IP address to the virtual routing interface.

 © 2012 Brocade Communications Systems, Inc.

If you configure an IP address on an individual port, you can configure Layer 3 interface parameters on that port. If you configure a virtual routing interface, you can configure Layer 3 interface parameters only on the virtual routing interface. This also applies to security features such as SYN-Guard and SYN-Defense.

The source-nat Parameter

Some configurations require use of the source-nat parameter. This parameter changes the source IP address in a packet from a client to a server. When the ServerIron and server are in different sub-nets, this parameter ensures that the client's request appears to the real server to have come from the ServerIron, and ensures that the server reply goes back through the ServerIron.

In software release 07.2.x, use of this parameter also requires that you use the **server source-ip** or **source-ip** command to configure an IP interface in the same sub-net as the server. In TrafficWorks 8.0, you do not need these commands. Instead, use the ip address command to configure an IP interface that is in the same sub-net as the server. Configure the interface on one of the following:

- The physical port connected to the server
- The virtual routing interface associated with the port-based VLAN that contains the port connected to the server

Disabling Layer 2 Switching

By default, ServerIron Chassis devices support Layer 2 switching. In Release 08.1.00R and later, if you want to disable Layer 2 switching, you can do so globally or on individual ports with the **route-only** command. This command is supported only on an interface facing a real server.

As a best practice, Brocade recommends you do not put **route-only** on the interfaces where servers are connected.

Configuring a Decnet Protocol VLAN

All ports will by default be assigned to the VLAN when initially created. VLAN Membership can be modified using the **dynamic**, **static**, or **exclude** commands.

To create a Decnet protocol VLAN on the ServerIron, enter commands such as the following:

```
ServerIron(config)# decnet-proto
ServerIron(config-decnet-proto)# static e 1/15 to 1/16
ServerIron(config-decnet-proto)# exclude e 1/1 to 1/14 e 1/18
```

This example creates a Decnet protocol VLAN with permanent port membership of 15 and 16 with port 17 as a dynamic member port (on module 1).

Syntax: [no] decnet-proto

Configuring an IP Interface

In Router (R) images, use the **ip address** command to configure an IP interface for use with IP forwarding. You must configure the IP interface on a virtual routing interface. You cannot configure the interface on a physical port. See **`router-interface`**.

To add an IP interface, enter commands such as the following:

```
ServerIron(config)# interface ve 1
ServerIron(config-vif-1)# ip address 10.10.10.1 255.255.255.0
```

The **interface ve 1** command changes the CLI to the configuration level for virtual routing interface 1. The **ip address** command adds an IP interface.

Syntax: [no] ip address | nat-address | standby-address <ip-addr> <ip-mask> I <ip-addr>/<mask-bits>

 © 2012 Brocade Communications Systems, Inc.

This command applies only to Layer 3 IP interfaces for use with IP forwarding.

The **address | nat-address | standby-address** parameter identifies the type of IP interface you are adding.

- The **address** parameter adds a standard IP interface. This option is applicable in most cases.
- The **nat-address** parameter applies to active-standby configurations. This parameter configures a shared IP interface for use with SLB source NAT. Enter the same command with the same IP address on each of the ServerIrons in the active-standby configuration. The address is active only on one ServerIron (the ServerIron that is currently active) at a time.

  NOTE: SLB source NAT is different from standard Network Address Translation (NAT).

- The **standby-address** parameter applies to active-standby configurations and allows both ServerIrons to share the same router interface. One of the ServerIrons actively supports the interface while the other ServerIron provides failover for the interface if the first ServerIron becomes unavailable. Real servers can use the shared interface as their default gateway. Enter the same command with the same IP address on each of the ServerIrons in the active-standby configuration. The address is active only on one ServerIron (the ServerIron that is currently active) at a time.

The <ip-addr> parameter specifies the IP address.

The <ip-mask> parameter specifies a class-based (or “Classical”) IP sub-net mask.

The <mask-bits> parameter specifies the number of significant bits in a Classless Interdomain Routing (CIDR) sub-net mask.

You can use either format to configure the interface. For example, both the following commands are valid and produce the same result:

- ip address 10.10.10.1 255.255.255.0
- ip address 10.10.10.1/24

Configuring an IP Filter

You can use IP filters (or ACLs) to selectively control SLB and TCS traffic. The filters or ACLs can match on source and destination IP address, network mask, and TCP/UDP port information.

All filters and ACLs are dynamic; they take place immediately for new connections and do not require a reboot of the ServerIron. New filters or ACLs do not affect existing connections.

Each filter or ACL provides one of the following actions:

- Permit
  - For SLB, permits access to a virtual server (identified by VIP) or to a specific TCP/UDP port on the virtual server.
  - For TCS, permits redirection of a client request to a cache server.
- Deny
  - For SLB, denies access to a virtual server (identified by VIP) or to a specific TCP/UDP port on the virtual server. The packet is dropped.
  - For TCS, denies access to the cache server and instead sends the request out to the Internet. The packet is not dropped.

By default, no filters or ACLs are configured on the ServerIron. All packets are implicitly permitted. However, as soon as you add a filter or ACL, all packets that do not match the filter or ACL are implicitly denied. This behavior ensures tighter control in filtered environments. To change this behavior so that all packets that do not match a filter are permitted instead of denied, configure the last filter (1024) or ACL to permit any traffic.

 © 2012 Brocade Communications Systems, Inc.

NOTE: To filter on Layer 2 traffic, you can configure Layer 2 MAC filters. See “MAC Filters” on page 2-14.

To set up IP filters to explicitly permit or deny access to specific TCP/UDP ports, use the **ip filter** command. When you configure this type of filter, you specify the virtual IP address (VIP) as the destination address for the filter, not the real server’s IP address.

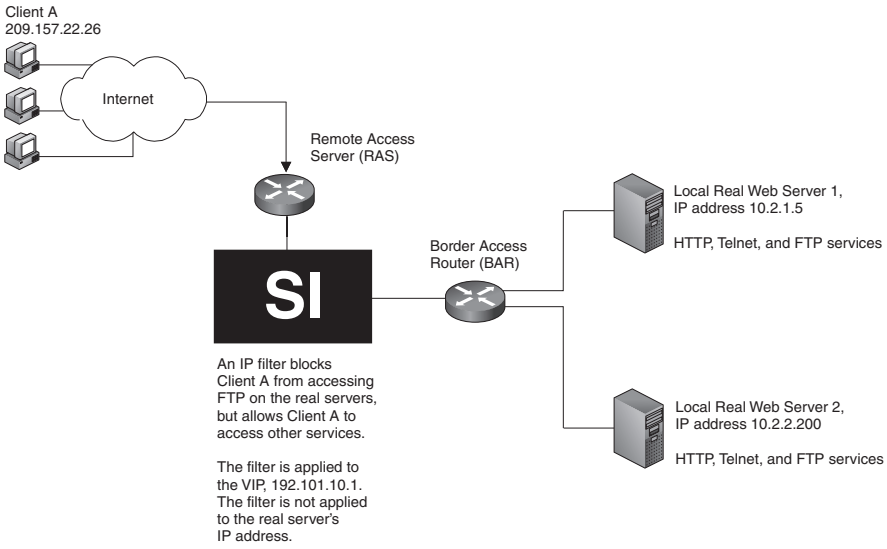
Syntax: [no] ip filter <filter-id> permit | deny <src-ip-addr> | any <src-mask> | any <dst-ip-addr> | any <dst-mask> | any <protocol> [<established> <operator> <port range>]

The items in brackets apply to TCP only.

SLB Example

Figure 2.3 shows an example of how you can use an IP filter in SLB. In this example, the administrator wants to block a specific client’s access to the FTP service on a VIP but permit access to the other services.

Figure 2.3 IP filter used to block client access to a TCP/UDP port



To configure an IP filter to block 209.157.22.26 from accessing FTP on 192.101.10.1:

```
ServerIron(config)#ip filter 1 deny 209.157.22.26 255.255.255.0 192.101.10.1
255.255.255.0 tcp eq ftp
```

You cannot use Layer 2 filters to filter Layer 4 information. To filter Layer 4 information, use IP access policies.

TCS Uses of Filters

You can use filters in TCS to control the following:

- Whether a specific request is sent to a cache server or forwarded to the Internet
- Whether content from specific sites is cached. You can even use policy-based cache switching to determine which cache servers receive content from specific sites.

NOTE: TCS filters never drop packets. Accept filters send packets to a cache server. Deny filters send packets to the Internet.

If you do not define any filters, the default action is permit. For TCS, the default action redirects all traffic to cache servers. However, when you define a filter, the ServerIron changes the default action to deny to ensure tighter control. If you still want the default action to be permit, you can define the last filter (1024) to permit all traffic.

 © 2012 Brocade Communications Systems, Inc.

Filters apply only to new connections. New filters do not affect existing connections.

You can turn off web caching for a certain range of source or destination addresses to allow filtering on an address basis using IP filters.

Policy-Based Cache Switching

The ServerIron TCS software allows you to configure IP filters to selectively cache or not cache content from specific web sites on specific cache servers. For example, suppose some of your cache servers come preconfigured with specific web pages and you want all updates to those pages to go only to the preconfigured caches. In this case, you can use policy-based cache switching along with IP filters to configure the ServerIron to send the content only to the specified cache servers.

You also can configure IP filters to prevent specific web sites from being cached on specific cache servers or all cache servers.

See “Policy-Based Caching” on page 15-37 for more information about this feature.

Setting the TTL

To set the maximum time that a packet will live on the network, enter the following command:

```
ServerIron(config)#ip ttl 25
```

Syntax: [no] ip ttl <hops>

<hops> is from 1 to 255. The default is 64.

Configuring an IP Protocol VLAN

When creating an IP protocol VLAN on a switch, all ports are dynamically assigned to the VLAN.

On a router, no ports are dynamically assigned to an IP protocol VLAN. VLAN port membership must be assigned using the **static** command, as shown in the example below. Because no dynamic port assignment is made for IP Protocol VLANs on a router, there is no need to exclude any ports, only specify membership with the **static** command.

An IP protocol and IP sub-net VLAN cannot operate simultaneously on a Brocade switch or router. This restriction is also true for IPX and IPX network VLANs. If you have previously defined an IP sub-net VLAN on the system, you need to delete it before an IP protocol VLAN can be created.

To assign ports 1, 2, 6 and 8 to an IP protocol VLAN, enter the following commands:

```
ServerIron(config)#ip-proto
ServerIron(config-ip-proto)#static e1 to 2 e6 e8
```

Syntax: [no] ip-proto

Configuring an IP Subnet Protocol VLAN

An IP sub-net protocol VLAN on a switch or router allows you to provide additional granularity than that of an IP protocol VLAN, by allowing broadcast domains to be partitioned by sub-net. As with the IP protocol VLAN, port membership can be modified using the **static** commands. In creating an IP sub-net VLAN, an IP address is used as an identifier.

When creating an IP sub-net VLAN on a switch, all ports are dynamically assigned to the VLAN.

On a router, no ports are dynamically assigned to an IP sub-net VLAN. VLAN port membership must be assigned using the static command, as shown in the example below. Because no dynamic port assignment is made for IP sub-net VLANs on a router, there is no need to exclude any ports, only specify membership with the **static** command.

 © 2012 Brocade Communications Systems, Inc.

NOTE: An IP Protocol and IP sub-net VLAN cannot operate simultaneously on a Brocade switch or router. This restriction is also true for IPX and IPX network VLANs. If you have previously defined an IP protocol VLAN on the system, you need to delete it before an IP sub-net VLAN can be created.

To create an IP sub-net of IP address 192.75.3.0 with permanent port membership of 1 and 2, enter the following commands:

```
BigIron(config)#ip-subnet 192.75.3.0 255.255.255.0
BigIron(config-ip-subnet)#static e1 to 2
BigIron(config-ip-subnet)#exit
```

Syntax: [**no**] **ip-subnet** <ip-addr> <ip-mask>

RIP

The ServerIron supports the following RIP versions:

- Version 1
- V1 compatible with V2
- Version 2 (the default)

You can configure the following parameters:

- Global parameters:
  - Administrative distance
  - Redistribution
  - Update interval
  - Learning of default route
  - Advertising and learning with specific neighbors
- Interface parameters:
  - RIP version
  - Metric
  - Learning of default route
  - Split horizon
  - Poison reverse
  - Advertising and learning of specific routes

For comprehensive information about the RIP features and how to configure them, see the "Configuring RIP" chapter in the *Foundry Enterprise Configuration and Management Guide*.

Enabling RIP

RIP is disabled by default. If you want the ServerIron to use RIP you must enable the protocol globally, then enable RIP on the virtual routing interface and specify the version (version 1 only, version 2 only, or version 1 compatible with version 2).

To globally start the RIP process, enter the following command:

```
ServerIron(config)#router rip
```

Syntax: [no] router rip

To enable RIP on the virtual routing interface and specify the RIP version, enter the following commands:

 © 2012 Brocade Communications Systems, Inc.

```
ServerIron(config-rip-router)#interface ve 1
ServerIron(config-vif-1)#ip rip v1-only
```

Syntax: [no] ip rip v1-only | v1-compatible-v2 | v2-only

RIP Timers

ServerIron Release 10.2.00 enhances the current functionality by providing support for RIP timers, such as update, aging, and garbage collection.

To configure RIP timers, use commands such as the following:

```
ServerIron(config) router rip
ServerIron (config-rip)# timers-basic 5 15 15
```

Syntax: [no] timers-basic <update-timer> <aging-timeout-interval> <garbage-collection-timer>

- The <update-timer> specifies how often RIP update messages are sent. You can specify from 1 - 1,000 seconds. The default is 30 seconds.
- The <aging-timeout-interval> specifies how long the Brocade device waits for a route update before declaring a route invalid. The value specified for the <aging-timeout-interval> should be at least three times the value specified for the <update-timer>. The <aging-timeout-interval> can be from 3 - 3,000 seconds. The default is 180 seconds.

The <garbage-collection-timer> specifies how long the Brocade device waits for a route update before removing the route from the RIP route table. The value specified for the <garbage-collection-timer> should be at least three times the value specified for the <update-timer>. The <garbage-collection-timer> can be from 3 - 3,000 seconds. The default is 120 seconds.

Redistributing IP Static Routes into RIP

By default, the software does not redistribute the IP static routes in the route table into RIP. To configure redistribution, perform the following tasks:

- Configure redistribution filters (optional). You can configure filters to permit or deny redistribution for a route based on the route's metric. You also can configure a filter to change the metric. You can configure up to 64 redistribution filters. The software uses the filters in ascending numerical order and immediately takes the action specified by the filter. Thus, if filter 1 denies redistribution of a given route, the software does not redistribute the route, regardless of whether a filter with a higher ID permits redistribution of that route.

  NOTE: The default redistribution action is permit, even after you configure and apply a permit or deny filter. To deny redistribution of specific routes, you must configure a deny filter.

  NOTE: The option to set the metric is not applicable to static routes.

- Enable redistribution.

  NOTE: If you plan to configure redistribution filters, do not enable redistribution until you have configured the filters.

When you enable redistribution, all IP static routes are redistributed by default. If you want to deny certain routes from being redistributed into RIP, configure deny filters for those routes before you enable redistribution. You can configure up to 64 RIP redistribution filters. They are applied in ascending numerical order.

NOTE: The default redistribution action is still permit, even after you configure and apply redistribution filters to the virtual routing interface. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (filter ID 64), then apply filters with lower filter IDs to allow specific routes.

To configure a redistribution filter, enter a command such as the following:

 © 2012 Brocade Communications Systems, Inc.

```
ServerIron(config-rip-router)# deny redistribute 1 static address 207.92.0.0
255.255.0.0
```

This command denies redistribution of all 207.92.x.x IP static routes.

Syntax: [no] permit | deny redistribute <filter-num> static address <ip-addr> <ip-mask> [match-metric <value> | set-metric <value>]

The <filter-num> specifies the redistribution filter ID. Specify a number from 1 – 64. The software uses the filters in ascending numerical order. Thus, if filter 1 denies a route from being redistributed, the software does not redistribute that route even if a filter with a higher ID permits redistribution of the route.

The **address** <ip-addr> <ip-mask> parameters apply redistribution to the specified network and sub-net address. Use 0 to specify "any". For example, "207.92.0.0 255.255.0.0" means "any 207.92.x.x sub-net". However, to specify any sub-net (all sub-nets match the filter), enter "address 255.255.255.255 255.255.255.255".

The **match-metric** <value> parameter applies redistribution to those routes with a specific metric value; possible values are from 1 – 15.

The **set-metric** <value> parameter sets the RIP metric value that will be applied to the routes imported into RIP.

NOTE: The **set-metric** parameter does not apply to static routes.

The following command denies redistribution of a 207.92.x.x IP static route only if the route's metric is 5.

```
ServerIron(config-rip-router)# deny redistribute 2 static address 207.92.0.0
255.255.0.0 match-metric 5
```

The following commands deny redistribution of all routes except routes for 10.10.10.x and 20.20.20.x:

```
ServerIron(config-rip-router)# deny redistribute 64 static address 255.255.255.255
255.255.255.255
ServerIron(config-rip-router)# permit redistribute 1 static address 10.10.10.0
255.255.255.0
ServerIron(config-rip-router)# permit redistribute 2 static address 20.20.20.0
255.255.255.0
```

Enabling Redistribution

After you configure redistribution parameters, you need to enable redistribution, by entering the following command:

```
ServerIron(config-rip-router)#redistribution
```

Syntax: redistribution

Denying Redistribution

You can configure a redistribution filter to deny redistribution for specific routes. By default, all routes are permitted to be redistributed

When you enable redistribution, all IP static routes are redistributed by default. If you want to deny certain routes from being redistributed into RIP, configure deny filters for those routes before you enable redistribution. You can configure up to 64 RIP redistribution filters. They are applied in ascending numerical order.

The default redistribution action is still permit, even after you configure and apply redistribution filters to the virtual routing interface. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (filter ID 64), then apply filters with lower filter IDs to allow specific routes.

To configure a redistribution filter, enter the following command:

```
ServerIron(config-rip-router)# deny redistribute 1 static address 207.92.0.0
255.255.0.0
```

This command denies redistribution of all 207.92.x.x IP static routes.

The following command denies redistribution of a 207.92.x.x IP static route only if the route's metric is 5.

 © 2012 Brocade Communications Systems, Inc.

```
ServerIron(config-rip-router)# deny redistribute 2 static address 207.92.0.0
255.255.0.0 match-metric 5
```

The following commands deny redistribution of all routes except routes for 10.10.10.x and 20.20.20.x:

```
ServerIron(config-rip-router)# deny redistribute 64 static address 255.255.255.255
255.255.255.255
ServerIron(config-rip-router)# permit redistribute 1 static address 10.10.10.0
255.255.255.0
ServerIron(config-rip-router)# permit redistribute 2 static address 20.20.20.0
255.255.255.0
```

Syntax: [no] deny redistribute <filter-num> static address <ip-addr> <ip-mask> [match-metric <value> | set-metric <value>]

The <filter-num> specifies the redistribution filter ID. Specify a number from 1 – 64. The software uses the filters in ascending numerical order. Thus, if filter 1 denies a route from being redistributed, the software does not redistribute that route even if a filter with a higher ID permits redistribution of the route.

The **address** <ip-addr> <ip-mask> parameters apply redistribution to the specified network and sub-net address. Use 0 to specify "any". For example, "207.92.0.0 255.255.0.0" means "any 207.92.x.x sub-net". However, to specify any sub-net (all sub-nets match the filter), enter "address 255.255.255.255 255.255.255.255".

The **match-metric** <value> parameter applies redistribution to those routes with a specific metric value; possible values are from 1 – 15.

The **set-metric** <value> parameter sets the RIP metric value that will be applied to the routes imported into RIP.

NOTE: The **set-metric** parameter does not apply to static routes.

Permitting Redistribution

You can configure a redistribution filter to permit redistribution for specific routes. When you enable redistribution, all IP static routes are redistributed by default. If you want to permit certain routes to be redistributed into RIP, configure permit filters for those routes before you enable redistribution.

The default redistribution action is permit, even after you configure and apply redistribution filters to the virtual routing interface. If you want to tightly control redistribution, apply a filter to deny all routes as the last filter (filter ID 64), then apply filters with lower filter IDs to allow specific routes.

To configure a redistribution filter, enter the following command:

```
ServerIron(config-rip-router)# permit redistribute 1 static address 207.92.0.0
255.255.0.0
```

This command permits redistribution of all 207.92.x.x IP static routes.

Syntax: permit redistribute

You can configure up to 64 RIP redistribution filters. They are applied in ascending numerical order. All routes are permitted to be redistributed.

Syntax: [no] permit redistribute <filter-num> static address <ip-addr> <ip-mask> [match-metric <value> | set-metric <value>]

The <filter-num> specifies the redistribution filter ID. Specify a number from 1 – 64. The software uses the filters in ascending numerical order. Thus, if filter 1 denies a route from being redistributed, the software does not redistribute that route even if a filter with a higher ID permits redistribution of the route.

The **address** <ip-addr> <ip-mask> parameters apply redistribution to the specified network and sub-net address. Use 0 to specify "any". For example, "207.92.0.0 255.255.0.0" means "any 207.92.x.x sub-net". However, to specify any sub-net (all sub-nets match the filter), enter "address 255.255.255.255 255.255.255.255".

The **match-metric** <value> parameter applies redistribution to those routes with a specific metric value; possible values are from 1 – 15.

The **set-metric** <value> parameter sets the RIP metric value that will be applied to the routes imported into RIP.

 © 2012 Brocade Communications Systems, Inc.

NOTE: The **set-metric** parameter does not apply to static routes.

Learning RIP Default Routes

By default, the software does not learn RIP default routes. To enable learning of RIP default routes, enter the following commands:

```
ServerIron(config)#interface ve 1
ServerIron(config-vif-1)#ip rip learn-default
```

Syntax: [no] ip rip learn-default

Enabling Poison Reverse or Split Horizon

RIP can use the following methods to prevent routing loops:

- Split horizon – The ServerIron does not advertise a route on the same interface as the one on which the ServerIron learned the route.
- Poison reverse – The ServerIron assigns a cost of 16 (“infinite” or “unreachable”) to a route before advertising it on the same interface as the one on which the ServerIron learned the route. This is the default.

These methods are in addition to RIP’s maximum valid route cost of 15.

To enable split horizon, enter the following commands:

```
ServerIron(config)#interface ve 1
ServerIron(config-vif-1)#no ip rip poison-reverse
```

Syntax: [no] ip rip poison-reverse

OSPF

The ServerIron supports OSPF RFC 1583 by default. Optionally, you can enable support for RFC 2178.

You can configure the following OSPF parameters:

- Global parameters:
  - Areas (standard, backbone, stub, and NSSA)
  - Area ranges
  - Virtual links
  - Default metric
  - Reference bandwidth for the default cost of OSPF interfaces
  - Route path load sharing
  - Default route origination
  - Shortest Path First (SPF) timers
  - External route summarization
  - Redistribution
  - Redistribution metric type
  - LSA pacing interval
  - OSPF traps
  - Database overflow interval

 © 2012 Brocade Communications Systems, Inc.

- Interface parameters:
  - Area membership
  - Authentication (simple password or MD5)
  - Link cost
  - Interface priority
  - Retransmit interval, transit delay, and dead interval

For information about the OSPF features and how to configure them, see the "Configuring OSPF" chapter in the *Foundry Enterprise Configuration and Management Guide*.

Dynamic Link Aggregation

The software supports the IEEE 802.3ad standard for link aggregation. This standard describes the Link Aggregation Control Protocol (LACP), a mechanism for allowing ports on both sides of a redundant link to configure themselves into a trunk link (aggregate link), without the need for manual configuration of the ports into trunk groups.

When you enable link aggregation on a group of Brocade ports, the Brocade ports can negotiate with the ports at the remote ends of the links to establish trunk groups.

Configuration Rules

- Use the link aggregation feature only if the device at the other end of the links you want to aggregate also supports IEEE 802.3ad link aggregation. Otherwise, you need to manually configure the trunk links.
- You cannot use 802.3ad link aggregation on a port configured as a member of a static trunk group.
- Link aggregation support is disabled by default. You can enable the feature on an individual port basis, in active or passive mode.
- Brocade recommends that you disable or remove the cables from the ports you plan to enable for dynamic link aggregation. Doing so prevents the possibility that LACP will use a partial configuration to talk to the other side of a link. A partial configuration does not cause errors, but does sometimes require LACP to be disabled and re-enabled on both sides of the link to ensure that a full configuration is used. It's easier to disable a port or remove its cable first. This applies both for active link aggregation and passive link aggregation.
- Active mode – When you enable a port for active link aggregation, the Brocade port can exchange standard LACP Protocol Data Unit (LACPDU) messages to negotiate trunk group configuration with the port on the other side of the link. In addition, the Brocade port actively sends LACPDU messages on the link to search for a link aggregation partner at the other end of the link, and can initiate an LACPDU exchange to negotiate link aggregation parameters with an appropriately configured remote port.
- Passive mode – When you enable a port for passive link aggregation, the Brocade port can exchange LACPDU messages with the port at the remote end of the link, but the Brocade port cannot search for a link aggregation port or initiate negotiation of an aggregate link. Thus, the port at the remote end of the link must initiate the LACPDU exchange.
- When the feature dynamically adds or changes a trunk group, the **show trunk** command displays the trunk as both configured and active. However, the **show running-config** or **write terminal** command does not contain a trunk command defining the new or changed trunk group.
- If link aggregation places a port into a trunk group as a secondary port, all configuration information except information related to link aggregation is removed from the port. For example, if port 1/3 has an IP interface, and the link aggregation feature port 1/3 into a trunk group consisting of ports 1/1 – 1/4, the IP interface is removed from the port.
- If you use this feature on a system running Router code that is running OSPF or BGP4, the feature causes these protocols to reset when a dynamic link change occurs. The reset includes ending and restarting

 © 2012 Brocade Communications Systems, Inc.

neighbor sessions with OSPF and BGP4 peers, and clearing and relearning dynamic route entries and forwarding cache entries. Although the reset causes a brief interruption, the protocols automatically resume normal operation.

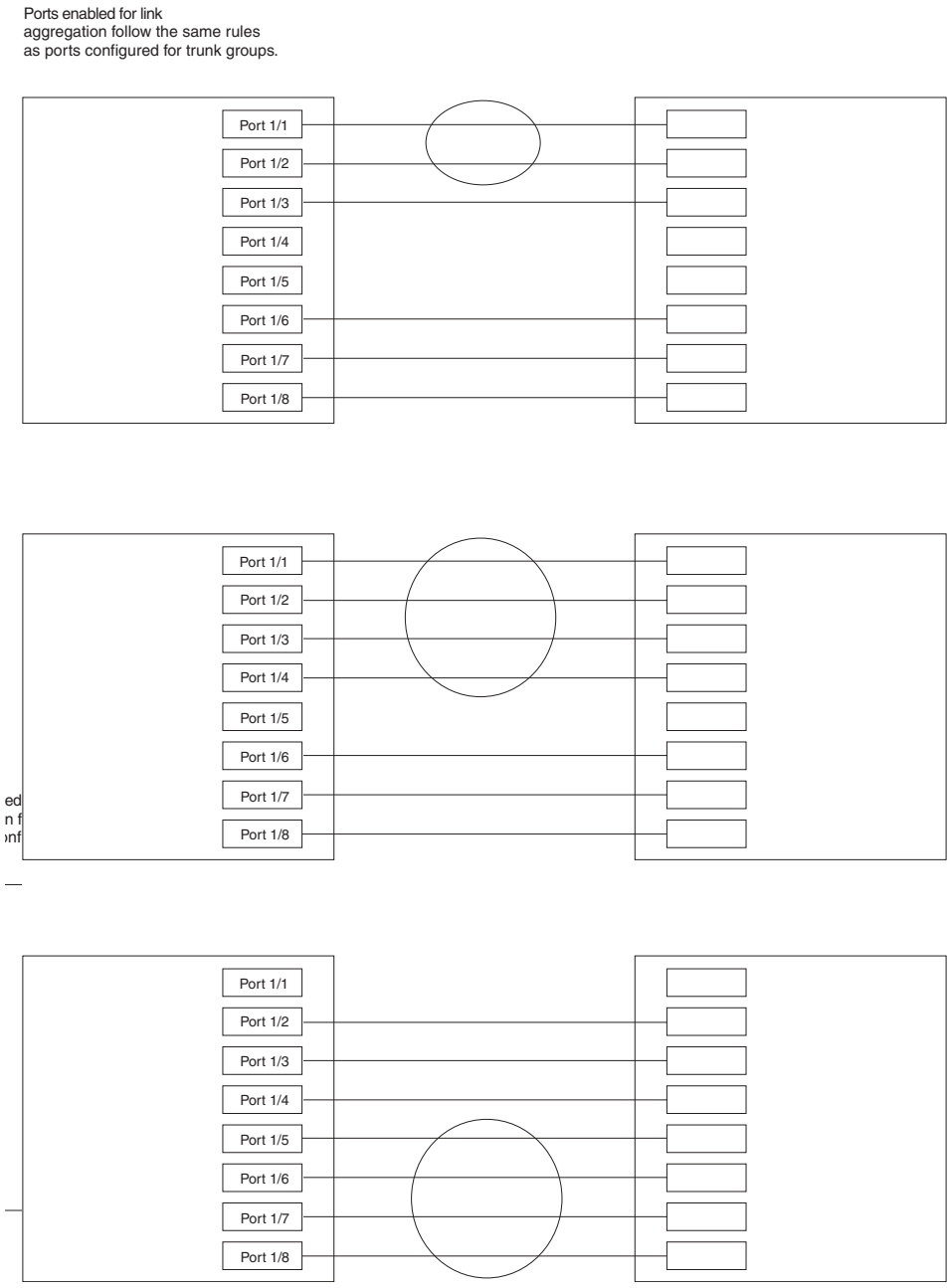
- If a device changes the number of ports in an active aggregate link, the Brocade device on the other end of the link tears down the link. Once the other device recovers, 802.3 can renegotiate the link without a mismatch.
- You can configure one or more parameters on the same command line, and you can enter the parameters in any order.

Valid Aggregate Links

Figure 2.4 on page 2-43 shows some examples of valid aggregate links.

 © 2012 Brocade Communications Systems, Inc.

Figure 2.4 Examples of valid aggregate links



In this example, assume that link aggregation is enabled on all of the links between the Brocade device on the left and the device on the right (which can be either a Brocade device or another vendor's device). Notice that some ports are not able to join an aggregate link even though link aggregation is enabled on them. The ports that are not members of aggregate links in this example are not following the configuration rules for trunk links on Brocade devices.

The Brocade rules apply to a Brocade device even if the device at the other end is from another vendor and uses different rules.

The link aggregation feature automates trunk configuration but can coexist with Brocade's trunk group feature. Link aggregation parameters do not interfere with trunk group parameters.

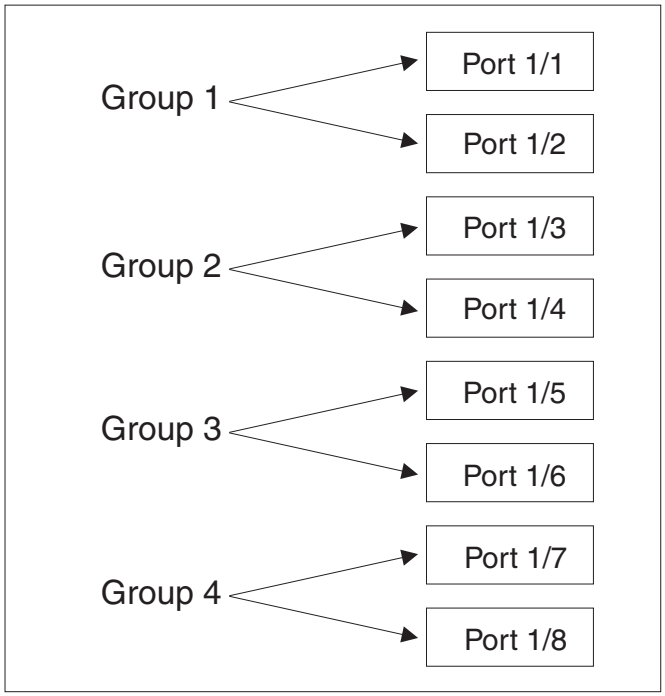
 © 2012 Brocade Communications Systems, Inc.

Flexible Trunk Eligibility

Flexible Trunk Eligibility increases the tolerance for down ports during link negotiation. In a valid trunk configuration (2-port, 4-port, or 8-port trunk starting on a valid primary port number) the device groups the device's ports into 2-port groups consisting of an odd-numbered port and the next even-numbered port. For example, ports 1/1 and 1/2 are a two-port group, as are ports 1/3 and 1/4, 9/1 and 9/10, and do on. If either of the ports in a two-port group is up, the device considers both ports to be eligible to be in an aggregate link.

Figure 2.5 shows an example of 2-port groups in a range of eight ports on which link aggregation is enabled. Based on the states of the ports, some or all of them will be eligible to be used in an aggregate link.

Figure 2.5 Two-port groups used to determine aggregation eligibility



© 2012 Brocade Communications Systems, Inc.

Table 2.7 shows examples of the ports from Figure 2.5 that will be eligible for an aggregate link based on individual port states.

Table 2.7: Port Eligibility for Link Aggregation

	Port Group 1		Port Group 2		Port Group 3		Port Group 4		Trunk Eligibility
	1/1	1/2	1/3	1/4	1/5	1/6	1/7	1/8	
Link State	Up	Up	Up	Up	Up	Up	Up	Up	8-port 1/1 – 1/8
	Up	Up	Up	Up	Up	Down	Up	Up	8-port 1/1 – 1/8
	Up	Up	Up	Up	Up	Down	Up	Down	8-port 1/1 – 1/8
	Up	Up	Up	Up	Down	Down	Down	Up	4-port 1/1 – 1/4
	Down	Down	Down	Up	Up	Up	Up	Up	4-port 1/5 – 1/8
	Up	Down	Down	Down	Up	Down	Down	Down	2-port 1/1 – 1/2

As shown in these examples, all or a subset of the ports within a port range will be eligible for formation into an aggregate link based on port states. Notice that the sets of ports that are eligible for the aggregate link must be valid static trunk configurations. For example, a 4-port link consisting of ports 1/4 – 1/7 is not valid because this port configuration is not valid for static trunk groups on the Brocade device.

Enabling Link Aggregation

By default, link aggregation is disabled on all ports.

NOTE: Configuration commands for link aggregation differ depending on whether you are using the default link aggregation key automatically assigned by the software, or if you are assigning a different, unique key or if you are assigning a key on a port on which link aggregation is already enabled. Follow the apropriate commands below.

Enabling Link Aggregation and Using the Default Key

Use this command to enable link aggregation on ports on which link aggregation has not been enabled if you want the software to assign a link aggregation key.

```
ServerIron(config)# interface ethernet 1/1
ServerIron(config-if-e1000-1/1)# link-aggregate active
ServerIron(config)# interface ethernet 1/2
ServerIron(config-if-e1000-1/2)# link-aggregate active
```

Syntax: [no] link-aggregate active | passive | off

Note that these ports will use the default key, since one has not been explicitly configured. Also, the commands in this example enable the active mode of link aggregation on ports 1/1 and 1/2. The ports can send and receive LACPDU messages.

 © 2012 Brocade Communications Systems, Inc.

Assigning a Unique Key and Enabling Link Aggregation

Use this command sequence to assign a link aggregation key on ports that do not have link aggregation enabled, and for all other link aggregation parameters (i.e., system priority, port priority, and link type).

```
ServerIron(config)# interface ethernet 1/1
ServerIron(config-if-e1000-1/1)# link-aggregate configure key 10000
ServerIron(config-if-e1000-1/1)# link-aggregate active
ServerIron(config)# interface ethernet 1/2
ServerIron(config-if-e1000-1/2)# link-aggregate configure key 10000
ServerIron(config-if-e1000-1/2)# link-aggregate active
```

The commands in this example assign the key 10000 and enable the active mode of link aggregation on ports 1/1 and 1/2. The ports can send and receive LACPDU messages.

NOTE: As shown in this example, when configuring a key, it is pertinent that you assign the key prior to enabling link aggregation.

The following commands enable passive link aggregation on ports 1/5 – 1/8:

```
ServerIron(config)# interface ethernet 1/5 to 1/8
ServerIron(config-mif-1/5-1/8)# link-aggregate passive
```

The commands in this example enable the passive mode of link aggregation on ports 1/5 – 1/8. These ports wait for the other end of the link to contact them. After this occurs, the ports can send and receive LACPDU messages.

To disable link aggregation on a port, enter a command such as the following:

```
ServerIron(config-if-e1000-1/8)# link-aggregate off
```

Syntax: [no] link-aggregate active | passive | off

Syntax: [no] link-aggregate configure [system-priority <num>] | [port-priority <num>] | [key <num>] | [type server | switch]

See “Link Aggregation Parameters” on page 2-47 for details on the parameters in the command.

Configuring Keys For Ports with Link Aggregation Enabled

NOTE: As shown in this command sequence, to change the key on ports that already have link aggregation enabled, you must first turn OFF link aggregation, configure the new key, then re-enable link aggregation.

```
ServerIron(config)# interface ethernet 1/1 to 1/4
ServerIron(config-mif-1/1-1/4)# link-aggregate off
ServerIron(config-mif-1/1-1/4)# link-aggregate configure key 10000
ServerIron(config-mif-1/1-1/4)# link-aggregate active
ServerIron(config-mif-1/1-1/4)# interface ethernet 3/5 to 3/8
ServerIron(config-mif-3/5-3/8)# link-aggregate off
ServerIron(config-mif-3/5-3/8)# link-aggregate configure key 10000
ServerIron(config-mif-3/5-3/8)# link-aggregate active
```

These commands change the key for ports 1/1 – 1/4 and 3/5 – 3/8 to 10000. Since all ports in an aggregate link must have the same key, the command in this example enables ports 1/1 – 1/4 and 3/5 – 3/8 to form a multi-slot aggregate link.

Syntax: [no] link-aggregate active | passive | off

Syntax: [no] link-aggregate configure [system-priority <num>] | [port-priority <num>] | [key <num>] | [type server | switch]

See “Link Aggregation Parameters” on page 2-47 for details on the parameters in the command.

 © 2012 Brocade Communications Systems, Inc.

Link Aggregation Parameters

You can change the settings for the following link aggregation parameters, on an individual port basis:

System Priority

The **system-priority** <num> parameter specifies the Brocade device's link aggregation priority. On links on which link aggregation is enabled, **system priority** specifies the Brocade device's link aggregation priority relative to the devices at the other ends. A higher value indicates a lower priority. You can specify a priority from 0 – 65535. The default is 1.

NOTE: If you are connecting the Brocade device to another vendor's device and the link aggregation feature is not working, set the system priority on the Brocade device to a lower priority (a higher priority value). In some cases, this change allows the link aggregation feature to operate successfully between the two devices.

Port Priority

The **port-priority <num>** parameter determines the active and standby links. When a group of ports is negotiating with a group of ports on another device to establish a trunk group, the Brocade port with the highest priority becomes the default active port. The other ports (with lower priorities) become standby ports in the trunk group. You can specify a priority from 0 – 65535. A higher value indicates a lower priority. The default is 1.

NOTE: This parameter is not supported in the current software release. The primary port in the port group becomes the default active port. The primary port is the lowest-numbered port in a valid trunk-port group.

Link Type

The **type server | switch** parameter specifies whether the port group is connected to a server (**server**) or to another networking device (**switch**). The default is **switch**.

NOTE: When you change the trunk hashing from server to switch, or switch to server, you must disable or enable the trunk.
For static trunk, execute the **trunk deploy** command after changing the hashing type.
For dynamic trunk (LACP), first execute the **no link-aggr active** command on trunk ports, change the hashing type, and then execute the **link-aggregation active** command.

Key

Every port that is 802.3ad-enabled has a key. The **key** <num> parameter identifies the group of ports that are eligible to be aggregated into a trunk group. Ports with the same key are called a key group and are eligible to be in the same trunk group.

When you enable link aggregation on a tagged or untagged port, Brocade's software assigns a default key to the port. The default key is based on the position of the port within an eight-port group (the maximum number of ports in a trunk group on a Layer 3 Switch). The software assigns the keys in ascending numerical order, beginning with key 0 for the first group of eight ports. For example, a 24-port module in chassis slot 1 contains keys 0, 1, and 2 by default. Ports 1/1 – 1/8 have key 0, ports 1/9 – 1/16 have key 1, and so on.

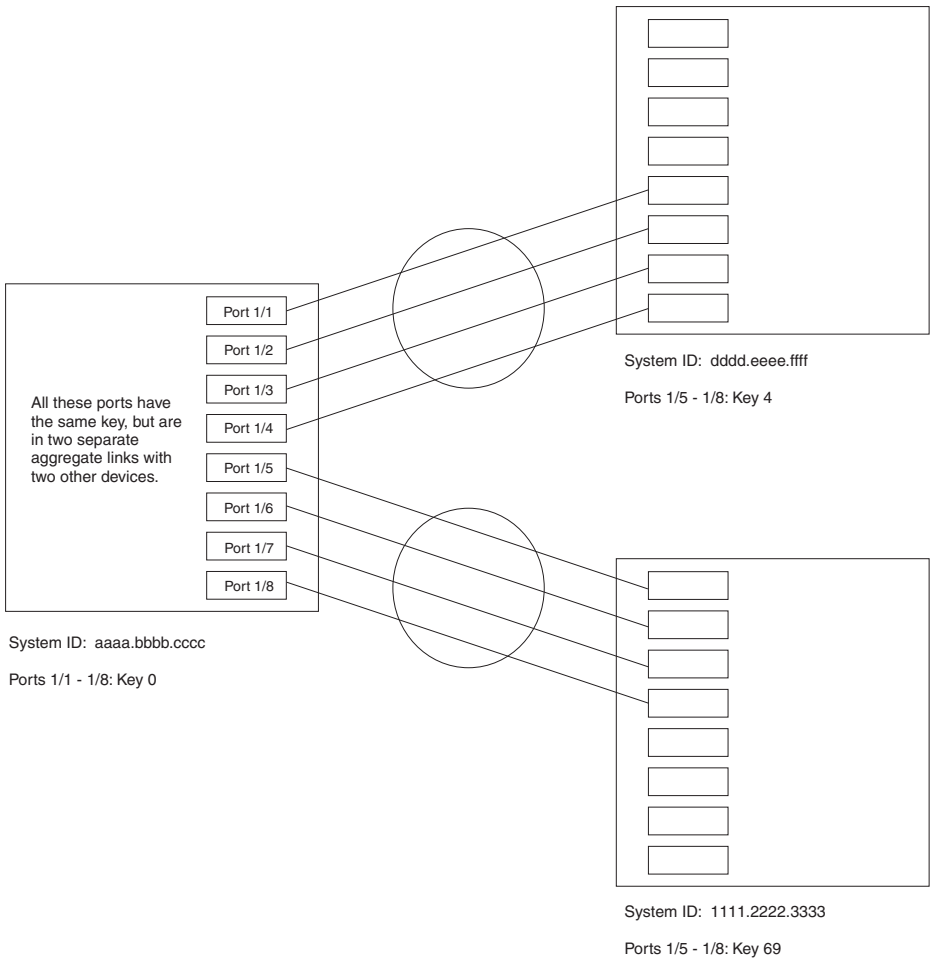
All ports within an aggregate link must have the same key. However, if the device has ports that are connected to two different devices, and the port groups allow the ports to form into separate aggregate links with the two devices, then each group of ports can have the same key while belonging to separate aggregate links with different devices.

NOTE: If you change the key for a port group, Brocade recommends that you use the value 10000 or higher, to avoid potential conflicts with dynamically created keys.

Figure 2.6 on page 2-48 shows an example of ports with the same key but are in different aggregate links.

 © 2012 Brocade Communications Systems, Inc.

Figure 2.6 Ports with the same key in different aggregate links

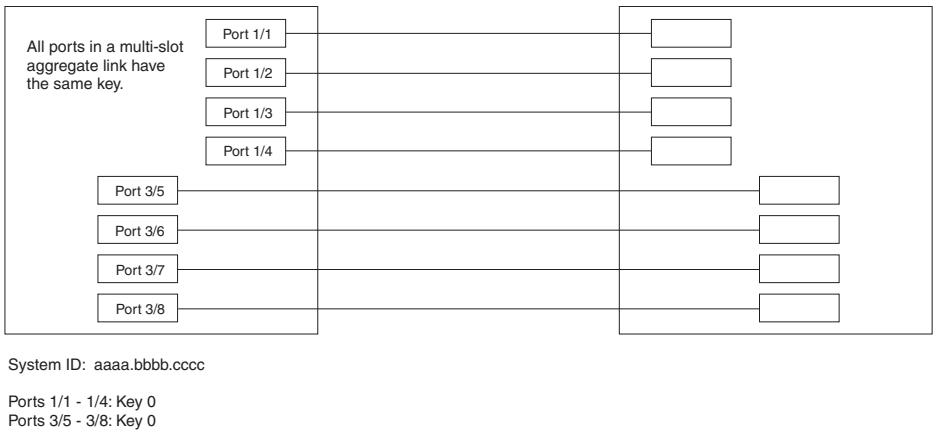

Port 1/1
Port 1/2
Port 1/3
Port 1/4
Port 1/5
Port 1/6
Port 1/7
Port 1/8
All these ports have the same key, but are in two separate aggregate links with two other devices.
System ID: aaaa.bbbb.cccc
Ports 1/1 - 1/8: Key 0
System ID: dddd.eeee.ffff
Ports 1/5 - 1/8: Key 4
System ID: 1111.2222.3333
Ports 1/5 - 1/8: Key 69

Notice that the keys between one device and another do not need to match. The only requirement for key matching is that all the ports within an aggregate link on a given device must have the same key.

Devices that support multi-slot trunk groups can form multi-slot aggregate links using link aggregation. However, the link aggregation keys for the groups of ports on each module must match. For example, if you want to allow link aggregation to form an aggregate link containing ports 1/1 – 1/4 and 3/5 – 3/8, you must change the link aggregation key on one or both groups of ports so that the key is the same on all eight ports. Figure 2.7 on page 2-49 shows an example.

 © 2012 Brocade Communications Systems, Inc.

Figure 2.7 Multi-slot aggregate link



By default, the device's ports are divided into 4-port groups. The software dynamically assigns a unique key to each 4-port group. If you need to divide a 4-port group into two 2-port groups, change the key in one of the groups so that the two 2-port groups have different keys. For example, if you plan to use ports 1/1 and 1/2 in VLAN 1, and ports 1/3 and 1/4 in VLAN 2, change the key for ports 1/3 and 1/4.

NOTE: If you change the key for a port group, Brocade recommends that you use the value 10000 or higher, to avoid potential conflicts with dynamically created keys.

About Blocked Ports

Brocade devices can block traffic on a port or shut down a port that is part of a trunk group or aggregate link for the following reasons:

- For the purpose of link aggregation, the ports on Brocade devices are grouped into pairs of two; one odd-numbered port, and the next even-numbered port. When you configure link aggregation on a port (for instance, on an odd-numbered port), this port will be blocked and unable to join a trunk group until you configure the adjacent port (the even-numbered port) as part of the aggregate link. When you configure both ports with link aggregation and assign both ports the same key, both ports are able to join a trunk group. Once the ports become part of a trunk group, they can transmit and receive LACP packets.

NOTE: Ports that are configured as part of an aggregate link must also have the same key. For more information about assigning keys, see the "Link Aggregation Parameters" on page 2-47

- When a port joins a trunk group and the port on the other end of the link shuts down or stops transmitting LACP packets, the Brocade device blocks the port. Depending on the timeout value set on the port, the link aggregation information expires.

If either of these events occur, the Brocade device shuts down the port and notifies all the upper layer protocols that the port is down.

Brocade devices can also block traffic on a port that is initially configured with link aggregation. The port is blocked until it joins a trunk group. In this case, traffic is blocked, but the port is still operational.

A port remains blocked until one of the following events occur:

- Link aggregation is enabled on the adjacent port (the paired port) and both ports have the same key
- LACP brings the port back up
- The port joins a trunk group

 © 2012 Brocade Communications Systems, Inc.

Displaying and Determining the Status of Aggregate Links

Use the **show link-aggregation** command to determine the operational status of ports associated with aggregate links.

To display the link aggregation information for a specific port, enter a command such as the following at any level of the CLI:

```
ServerIron(config-mif-1/1-1/8)# show link-aggregation ethernet 1/1
System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp] [Ope]
1/1        0        0       0   No    L   No   No   No   No   No   No    Ope
```

The command in this example shows the link aggregation information for port 1/1.

To display the link aggregation information for all ports on which link aggregation is enabled, enter the following command at any level of the CLI:

```
ServerIron(config)# show link-aggregation
System ID: 00e0.52a9.bb00
Port  [Sys P] [Port P] [ Key ] [Act][Tio][Agg][Syn][Col][Dis][Def][Exp][Ope]
1/1        1        1       0   No    L   Agg  Syn  No   No   Def  Exp  Ope
1/2        1        1       0   No    L   Agg  Syn  No   No   Def  Exp  Ina
1/3        1        1       0   No    L   Agg  Syn  No   No   Def  Exp  Ina
1/4        1        1       0   No    L   Agg  Syn  No   No   Def  Exp  Blo
1/5        1        1       1   No    L   Agg  No   No   No   Def  Exp  Ope
1/6        1        1       1   No    L   Agg  No   No   No   Def  Exp  Ope
1/7        1        1       1   No    L   Agg  No   No   No   Def  Exp  Dwn
1/8        1        1       1   No    L   Agg  No   No   No   Def  Exp  Dwn
```

Syntax: show link-aggregation [ethernet <portnum>]

Use **ethernet <portnum>** to display link-aggregation information for a specific port.

NOTE: Ports that are configured as part of an aggregate link must also have the same key. For more information about assigning keys, see the section titled ““Link Aggregation Parameters” on page 2-47.

The **show link aggregation** command shows the following information.

Table 2.8: CLI Display of Link Aggregation Information

This Field...	Displays...
System ID	Lists the base MAC address of the device. This is also the MAC address of port 1 (or 1/1).
Port	Lists the port number.
Sys P	Lists the system priority configured for this port.
Port P	Lists the port’s link aggregation priority.
Key	Lists the link aggregation key.

 © 2012 Brocade Communications Systems, Inc.

Table 2.8: CLI Display of Link Aggregation Information (Continued)

This Field...	Displays...
Act	Indicates the link aggregation mode, which can be one of the following: • No – The mode is passive or link aggregation is disabled (off) on the port. If link aggregation is enabled (and the mode is passive), the port can send and receive LACPDU messages to participate in negotiation of an aggregate link initiated by another port, but cannot search for a link aggregation port or initiate negotiation of an aggregate link. • Yes – The mode is active. The port can send and receive LACPDU messages.
Tio	Indicates the timeout value of the port. The timeout value can be one of the following: • L – Long. The trunk group has already been formed and the port is therefore using a longer message timeout for the LACPDU messages exchanged with the remote port. Typically, these messages are used as confirmation of the health of the aggregate link. • S – Short. The port has just started the LACPDU message exchange process with the port at the other end of the link. The S timeout value also can mean that the link aggregation information received from the remote port has expired and the ports are starting a new information exchange.
Agg	Indicates the link aggregation state of the port. The state can be one of the following: • Agg – Link aggregation is enabled on the port. • No – Link aggregation is disabled on the port.
Syn	Indicates the synchronization state of the port. The state can be one of the following: • No – The port is out of sync with the remote port. The port does not understand the status of the LACPDU process and is not prepared to enter a trunk link. • Syn – The port is in sync with the remote port. The port understands the status of the LACPDU message exchange process, and therefore knows the trunk group to which it belongs, the link aggregation state of the remote port, and so on.
Col	Indicates the collection state of the port, which determines whether the port is ready to send traffic over the trunk link. • Col – The port is ready to send traffic over the trunk link. • No – The port is not ready to send traffic over the trunk link.

 © 2012 Brocade Communications Systems, Inc.

Table 2.8: CLI Display of Link Aggregation Information (Continued)

This Field...	Displays...
Dis	Indicates the distribution state of the port, which determines whether the port is ready to receive traffic over the trunk link. • Dis – The port is ready to receive traffic over the trunk link. • No – The port is not ready to receive traffic over the trunk link.
Def	Indicates whether the port is using default link aggregation values. The port uses default values if it has not received link aggregation information through LACP from the port at the remote end of the link. This field can have one of the following values: • Def – The port has not received link aggregation values from the port at the other end of the link and is therefore using its default link aggregation LACP settings. • No – The port has received link aggregation information from the port at the other end of the link and is using the settings negotiated with that port.
Exp	Indicates whether the negotiated link aggregation settings have expired. The settings expire if the port does not receive an LACPDU message from the port at the other end of the link before the message timer expires. This field can have one of the following values: • Exp – The link aggregation settings this port negotiated with the port at the other end of the link have expired. The port is now using its default link aggregation settings. • No – The link aggregation values that this port negotiated with the port at the other end of the link have not expired, so the port is still using the negotiated settings.
Ope	• Ope (operational) - The port is operating normally. • Ina (inactive) - The port is inactive because the port on the other side of the link is down or has stopped transmitting LACP packets. • Blo (blocked) - The port is blocked because the adjacent port is not configured with link aggregation or because it is not able to join a trunk group. To unblock the port and bring it to an operational state, enable link aggregation on the adjacent port and ensure that the ports have the same key.

NOTE: Use the **show trunk** command to determine the status of LACP.

Clearing the Negotiated Link Aggregations

When a group of ports negotiates a trunk group configuration, the software stores the negotiated configuration in a table. You can clear the negotiated link aggregation configurations from the software. When you clear the information, the software does not remove link aggregation parameter settings you have configured. Only the configuration information negotiated using LACP is removed.

NOTE: The software automatically updates the link aggregation configuration based on LACPDU messages. However, clearing the link aggregation information can be useful if you are troubleshooting a configuration.

 © 2012 Brocade Communications Systems, Inc.

To clear the link aggregation information, enter the following command at the Privileged EXEC level of the CLI:

```
ServerIron# clear link-aggregate
```

Syntax: clear link-aggregate

© 2012 Brocade Communications Systems, Inc.

 © 2012 Brocade Communications Systems, Inc.